

Adaptive Incremental Mixture Markov chain Monte Carlo

Florian Maire, Nial Friel

School of Mathematics and Statistics, University College Dublin
and Insight Centre for Data Analytics

Antonietta Mira

Interdisciplinary Institute of Data Science, Università della Svizzera italiana
and DISAT, Università dell'Insubria

Adrian E. Raftery

Department of Statistics, University of Washington

April 28, 2016

Abstract

We propose Adaptive Incremental Mixture Markov chain Monte Carlo (AIMM), a novel approach to sample from challenging probability distributions defined on a general state-space. Typically, adaptive MCMC methods recursively update a parametric proposal kernel with a global rule; by contrast AIMM locally adapts a non-parametric kernel. AIMM is based on an independent Metropolis-Hastings proposal distribution which takes the form of a finite mixture of Gaussian distributions. Central to this approach is the idea that the proposal distribution adapts to the target by locally adding a mixture component when the discrepancy between the proposal mixture and the target is deemed to be too large. As a result, the number of components in the mixture proposal is not fixed in advance. Theoretically we prove that AIMM converges to the correct target distribution. We also illustrate that it performs well in practice in a variety of challenging situations, including high-dimensional and multimodal target distributions.

Keywords: Adaptive MCMC, Independence Sampler, Importance weight, Local adaptation, Bayesian inference.

1 Introduction

We consider the problem of sampling from a target distribution defined on a general state space. While standard simulation methods such as the Metropolis–Hastings algorithm (Metropolis et al., 1953; Hastings, 1970) and its many variants have been extensively studied, they can be inefficient in sampling from complex distributions such as those that arise in modern applications. For example, the practitioner is often faced with the issue of sampling from distributions which contain some or all of the following: multimodality, sparse high density regions, heavy tails, and high-dimensional support. In these cases, standard Markov chain Monte Carlo (MCMC) methods often have difficulties, leading to long mixing times and large asymptotic variances. The adaptive MCMC framework originally developed by Gilks and Wild (1992), Gilks et al. (1998), Haario et al. (1999) and Roberts and Rosenthal (2007) can help overcome these problems. Adaptive MCMC methods improve the convergence of the chain by tuning its transition kernel on the fly using the knowledge of the past trajectory of the process. This learning process causes a loss of the Markovian property, but for convenience we will still refer to the resulting stochastic process as a “Markov chain”.

Most of the adaptive MCMC literature to date has focused on updating an initial parametric proposal distribution. For example, the Adaptive Metropolis–Hastings algorithm (Haario et al., 1999, 2001), hereafter referred to as AMH, adapts the covariance matrix of a Gaussian proposal kernel, used in a Random Walk Metropolis–Hastings algorithm. The Adaptive Gaussian Mixtures algorithm (Giordani and Kohn, 2010; Luengo and Martino, 2013), hereafter referred to as AGM, adapts a mixture of Gaussian distributions, used as the proposal in an Independent Metropolis–Hastings algorithm.

When knowledge on the target distribution is limited, the assumption that the proposal kernel is restricted to a specific parametric family may lead to suboptimal performance. Moreover, a practitioner using these methods must choose, sometimes arbitrarily, (i) a parametric family and (ii) an initial set of parameters to start the sampler. However, poor choices of (i) and (ii) may constrain the adaptation and result in slow convergence.

In this paper, we introduce a novel adaptive MCMC method, called Adaptive Incremental Mixture Markov chain Monte Carlo (AIMM). This algorithm belongs to the general Adaptive Independent Metropolis class of methods, developed in Holden et al. (2009), in that AIMM

adapts an independence proposal density. However here the adaptation process is quite different from others previously explored in the literature. Although our objective remains to reduce the discrepancy between the proposal and the target distribution along the chain, AIMM proceeds without any parameter updating scheme. The idea is instead to add a *local* probability mass to the current proposal kernel, when a large discrepancy area between the target and the proposal is encountered. The local probability mass is added through a Gaussian kernel located in the region that is not sufficiently supported by the current proposal. The decision to increment the proposal kernel is based on the importance weight function that is implicitly computed in the Metropolis acceptance ratio. We stress that, although seemingly similar to Giordani and Kohn (2010) and Luengo and Martino (2013), our adaptation scheme is local and non-parametric, a subtle difference that has important theoretical and practical consequences. In particular, in contrast to AGM, the approach which we develop does not assume a fixed number of mixture components.

Proving the ergodicity of adaptive MCMC methods is often made easier by expressing the adaptation as a stochastic approximation of the proposal kernel parameters (Andrieu and Moulines, 2006). AIMM cannot be analysed in this framework as the adaptation does not proceed with a parameter update step. Instead, we prove that the Kullback–Leibler divergence, between the target distribution and the independence proposal, decreases along the chain, with increasing probability, an argument that is then used to show that the convergence conditions of Roberts and Rosenthal (2007) are satisfied.

The paper is organized as follows. We start in Section 2 with a pedagogical example which shows how AIMM succeeds in addressing the pitfalls of some adaptive MCMC methods. In Section 3, we formally present AIMM, and study its theoretical properties in Section 4. Section 5 illustrates the performance of AIMM on three different challenging target distributions in high dimensions, involving two heavy tailed distributions and a bimodal distribution. AIMM is also compared with other adaptive and non-adaptive MCMC methods. In Section 6 we discuss the connections with importance sampling methods, particularly Incremental Mixture of Importance Sampling (Raftery and Bao, 2010), from which AIMM takes inspiration.

2 Introductory example

We first illustrate some of the potential shortcomings of the adaptive methods mentioned in the introduction and outline how AIMM addresses them. We consider the following pedagogical example where the objective is to sample efficiently from a one-dimensional target distribution. Appendix E.1 gives more details of the various algorithms which we compare with AIMM and specifies the parameters of these algorithms.

Example 1. *Consider the target distribution*

$$\pi_1 = (1/4)\mathcal{N}(-10, 1) + (1/2)\mathcal{N}(0, .1) + (1/4)\mathcal{N}(10, 1) .$$

For this type of target distribution it is known that Adaptive Metropolis–Hastings (AMH) (Haario et al., 2001) mixes poorly since the three modes of π_1 are far apart, a problem faced by many non-independent Metropolis algorithms. Thus an adaptive independence sampler such as the Adaptive Gaussian Mixture (AGM) (Luengo and Martino, 2013) is expected to be more efficient. AGM uses the history of the chain to adapt the parameters of a mixture of Gaussians on the fly to match the target distribution. For AGM, two families of proposals are considered: the set of mixtures of two and three Gaussians, referred to as AGM–2 and AGM–3, respectively. By contrast, our method, Adaptive Incremental Mixture MCMC (AIMM) offers more flexibility in terms of the specification of the proposal distributions and in particular does not set a fixed number of mixture components.

We are particularly interested in studying the tradeoff between the convergence rate of the Markov chain and the asymptotic variance of the MCMC estimators, which are known to be difficult to control simultaneously (Mira, 2001) Table 1 gives information about the asymptotic variance, through the Effective Sample Size (ESS), and about the convergence of the chain, through the estimation of the tail-event probability $\pi_2(X > 5)$. Further details of these performance measures are given in Appendix E.2.

The ability of AMH to explore the state space appears limited as it regularly fails to visit the three modes in their correct proportions after 20,000 iterations (see the mean and the variance estimate of $\pi_2(X > 5)$ in Table 1). The efficiency of AGM depends strongly on the number of mixture components of the proposal family. In fact AGM–2 is comparable

Table 1: Quantitative Statistics for π_1 : Mean and variance of the Effective Sample Size (ESS) (the larger the better), mean and variance of the estimate of $\pi_1(X > 5)$ (theoretical value is $\pi_1(X > 5) = 1/4$) obtained through 100 independent runs of the four methods, each of 20,000 iterations.

π_2	ESS	$\pi_1(X > 5)$
AMH	(.08, .004)	(.284, 6×10^{-3})
AGM-2	(.12, .001)	(.245, 1×10^{-3})
AGM-3	(.51, .091)	(.256, 7×10^{-4})
AIMM	(.47, .004)	(.250, 5×10^{-5})

to AMH in terms of the average ESS, indicating that the adaptation has failed in most runs. AIMM and AGM-3 both achieve a similar exploration of the state space and AGM-3 offers a slightly better ESS. An animation corresponding to this toy example can be found at http://mathsci.ucd.ie/~fmaire/AIMM/toy_example.html.

From this example we conclude that AGM can be extremely efficient provided that some initial knowledge of the target distribution is available, for example, the number of modes, the location of the large density regions and so on. Otherwise, the inference can be jeopardized if a mismatch between the family of proposal distributions and the target occurs. Since this misspecification is typical in real world models where one encounters high-dimensional, multi-modal distributions and other challenging topologies, it leads one to question the efficiency of these types of samplers. On the other hand, AMH seems more robust to *a priori* lack of knowledge of the target, but the quality of the mixing of the chain remains a potential issue, especially when high density regions are disjoint.

Initiated with an naive independence proposal, see Section 3.4, AIMM adaptively builds a proposal that fits the target by adding probability mass to locations where the proposal is not well supported relatively to the target. As a result, very little, if any, information regarding the target distribution is needed, hence making AIMM robust to those realistic situations. Extensive experimentation in Section 5 confirms this point.

3 Adaptive Incremental MCMC

We consider target distributions π defined on a measurable space $(\mathsf{X}, \mathcal{X})$ where X is an open subset of \mathbb{R}^d ($d > 0$) and \mathcal{X} is the Borel σ -algebra of X . Unless otherwise stated, the distributions we consider are dominated by Lebesgue measure and we therefore denote the distribution and the density function (w.r.t Lebesgue measure) with the same symbol. Let $\{X_n, n \in \mathbb{N}\}$ be the sample path of a Markov chain produced by the Adaptive Incremental Mixture Markov chain Monte Carlo algorithm (AIMM). In this section, we introduce the AIMM transition kernel, defined as $K_n(X_n; A) = \Pr(X_{n+1} \in A | X_n)$, for any $A \in \mathcal{X}$.

3.1 Transition kernel

AIMM belongs to the general class of Adaptive Independent Metropolis algorithms originally introduced by Holden et al. (2009). It uses the past history of the Markov chain to refine a sequence of independence proposals $\{Q_n, n \in \mathbb{N}\}$ such that K_n is the standard Metropolis–Hastings (M–H) kernel with independence proposal Q_n and target π . For any $(x, A) \in (\mathsf{X}, \mathcal{X})$, K_n is defined by

$$K_n(x; A) = \int_A Q_n(\mathrm{d}x') \alpha_n(x, x') + \delta_x(A) \int_{\mathsf{X}} Q_n(\mathrm{d}x') (1 - \alpha_n(x, x')) . \quad (1)$$

In (1), α_n denotes the usual M–H acceptance probability for independence samplers, namely

$$\alpha_n(x, x') = 1 \wedge \frac{W_n(x')}{W_n(x)} , \quad (2)$$

where $W_n = \pi/Q_n$ is the importance weight function defined on X . Central to our approach is the idea that the discrepancy between π and Q_n , as measured by W_n , can be exploited in order to improve the independence proposal adaptively. This has the advantage that W_n is computed as a matter of course in the M–H acceptance probability (2).

We assume that available knowledge about π allows one to construct an initial proposal kernel Q_0 , from which it is straightforward to sample. When π is a posterior distribution, a default choice for Q_0 could be the prior distribution. For all $n \geq 0$, the proposal kernel at iteration n is defined by

$$Q_n = \omega_n Q_0 + (1 - \omega_n) \sum_{\ell=1}^{M_n} \beta_\ell \phi_\ell \Big/ \sum_{\ell=1}^{M_n} \beta_\ell , \quad (3)$$

where:

- $M_n \in \mathbb{N}$ is the number of times Q_n has been incremented up to the n -th transition ($M_n < n$ and $M_1 = 0$),
- $\omega_n = (1 + \kappa M_n)^{-1}$ is the probability of proposing from Q_0 at iteration n , ($\kappa \in (0, 1)$),
- $\{\phi_1, \dots, \phi_{M_n}\}$ is the collection of Gaussian components created by AIMM up to the n -th transition,
- $\{\beta_1, \dots, \beta_{M_n}\}$ is the set of unnormalized weights attached to the collection of Gaussian components.

Assigning a different weight to Q_0 at iteration n is motivated by the defensive mixture method (Hesterberg, 1995). Including Q_0 ensures that the importance weight function W_n is uniformly bounded. In what follows, we denote by $\{\tilde{X}_n, n \in \mathbb{N}\}$ the sequence of random variables consisting of the proposed states of the Markov chain, that is, $\tilde{X}_{n+1} \sim Q_n$.

3.2 Incremental process

The incremental process is driven by the random sequence $\{W_n(\tilde{X}_{n+1}), n \in \mathbb{N}\}$, which monitors the discrepancy between π and Q_n at the proposed state of the Markov chain \tilde{X}_{n+1} . Let W^* ($W^* > 0$) be a user-defined parameter such that the proposal kernel Q_n increments if the event \mathcal{E}_n

$$\mathcal{E}_n = \left\{ W_n(\tilde{X}_{n+1}) > W^*, \tilde{X}_{n+1} \sim Q_n \right\}$$

occurs. The specification of W^* involves a tradeoff between the asymptotic approximation of π by Q_0 and the computational efficiency of the algorithm:

- When $W^* \ll 1$, any negligible mismatch between Q_n and π will be corrected by the addition of a new component to the mixture proposal. In this case, one would expect that a large number of mixture components will be generated, resulting in a large computational burden.
- When $W^* \gg 1$, the mixture rarely increments ($Q_n \approx Q_0$) and the chain will rarely adapt. It may therefore suffer from bad mixing and poor exploration of π .

Upon the occurrence of \mathcal{E}_n a new component ϕ_{M_n+1} is created. It is set as a Gaussian distribution centered at $\mu_{M_n+1} = \tilde{X}_{n+1}$ and with covariance matrix Σ_{M_n+1} defined by

$$\Sigma_{M_n+1} = \text{Cov} \left\{ \mathfrak{N}(\tilde{X}_{n+1} | X_1, \dots, X_n) \right\}, \quad (4)$$

where for a set of vectors V , $\text{Cov}(V)$ denotes the empirical covariance matrix of V . In (4), $\mathfrak{N}(\tilde{X}_{n+1} | X_1, \dots, X_n) \subseteq (X_1, \dots, X_n)$ is a neighborhood of \tilde{X}_{n+1} defined as

$$\left\{ X_i \in (X_1, \dots, X_n), D_M(X_i, \tilde{X}_{n+1}) \leq \tau \rho_n \pi(\tilde{X}_{n+1}) \right\}, \quad (5)$$

where $\tau \in (0, 1)$ is a user-defined parameter controlling the neighborhood range and ρ_n the number of accepted proposals up to iteration n . In (5), $D_M(X_i, \tilde{X}_{n+1})$ denotes the Mahalanobis distance between X_i and \tilde{X}_{n+1} which relies on a covariance matrix Σ_0 . When π is a posterior distribution, Σ_0 could be, for example, the prior covariance matrix. The rationale behind the upper bound in (5) is that the neighborhood of a state \tilde{X}_{n+1} should be made larger if the target has been reasonably well explored by the Markov chain in the vicinity of \tilde{X}_{n+1} . This is expected to be positively correlated with $\pi(\tilde{X}_{n+1})$ and ρ_n . Finally, a weight is attached to the new component ϕ_{M_n+1} . It is proportional to

$$\beta_{M_n+1} = \pi(\tilde{X}_{n+1})^\gamma, \quad \gamma \in (0, 1).$$

The parameter $\gamma \in (0, 1)$ is user-defined and controls the tradeoff between the asymptotic variance and the convergence of the Markov chain. When $\gamma \nearrow 1$, components in high target probability areas are given higher weights and the acceptance rate is therefore expected to be higher as less hazardous moves in the low density regions are proposed. In contrast, when $\gamma \searrow 0$, all the components have the same weight, a setup that will foster a faster exploration of the state space.

3.3 AIMM

Algorithm 1 summarizes AIMM. Note that during an initial phase consisting of N_0 iterations, no adaptation is made. This serves the purpose of gathering sample points required to produce the first increment.

Algorithm 1 Adaptive Incremental Mixture MCMC

1: **Input:** user-defined parameters: $W^*, \gamma, \tau, \kappa, N_0$

a proposal distribution: Q_0

2: **Initialize:** $X_0 \sim Q_0$, $W_0 = \pi/Q_0$, $M_0 = 0$ and $\omega_0 = 1$

3: **for** $n = 0, 1, \dots$ **do**

4: Propose $\tilde{X}_{n+1} \sim Q_n = \omega_n Q_0 + (1 - \omega_n) \sum_{\ell=1}^{M_n} \beta_\ell \phi_\ell / \sum_{\ell=1}^{M_n} \beta_\ell$

5: Draw $U_n \sim \text{Unif}(0, 1)$ and set

$$\begin{cases} X_{n+1} = \tilde{X}_{n+1} & \text{if } U_n \leq \left(1 \wedge W_n(\tilde{X}_{n+1})/W_n(X_n)\right) \\ X_{n+1} = X_n & \text{otherwise} \end{cases}$$

6: **if** $W_n(\tilde{X}_{n+1}) > W^*$ and $n > N_0$ **then**

7: Update $M_{n+1} = M_n + 1$ and $\omega_{n+1} = (1 + \kappa M_{n+1})^{-1}$

8: Increment the proposal kernel with the new component

$$\phi_{M_{n+1}} = \mathcal{N}\left(\tilde{X}_{n+1}, \text{Cov}(\mathfrak{N}(\tilde{X}_{n+1} | X_1, \dots, X_n))\right)$$

weighted by $\beta_{M_{n+1}} = \pi(\tilde{X}_{n+1})^\gamma$

9: Update $W_{n+1} = \pi/Q_{n+1}$

10: **else**

11: Update $W_{n+1} = W_n$, $M_{n+1} = M_n$ and $\omega_{n+1} = \omega_n$

12: **end if**

13: **end for**

14: **Output:** $\{X_n, n \in \mathbb{N}\}$, $\{Q_n, n \in \mathbb{N}\}$

3.4 Example 1 (ctd.)

For the toy example in Section 2 we used the following parameters:

$$W^* = 1, \gamma = 0.5, \tau = 0.5, \kappa = 0.1, N_0 = 1,000.$$

Figure 1 shows the different states of the Markov chain sampled using $Q_0 = \mathcal{N}(0, 10)$ before the first increment takes place. The proposed state \tilde{X}_{n+1} activates the increment process when the condition $\{W_n(\tilde{X}_{n+1}) > W^*\}$ is satisfied for the first time after the initial phase ($N_0 = 1,000$) is completed. At \tilde{X}_{n+1} there is a large discrepancy between Q_0 and π_1 . The neighborhood of \tilde{X}_{n+1} is identified and in turn allows us to define ϕ_1 , the first component of the incremental mixture.

The top panel of Figure 2 shows the sequence of proposal distributions up to $n = 20,000$ iterations. AIMM has incremented the proposal 15 times. The first proposals are slightly bumpy and as more samples become available, the proposal appears to adhere better and better to the target distribution. Thus, even without any information about π_1 , AIMM is able to increment its proposal so that the discrepancy between Q_n and π_1 vanishes. This is confirmed by Figure 3 which reports the acceptance rate (averaged on a moving window) and the number of components in the mixture throughout the algorithm. Finally, the proposal kernels obtained after 20,000 iterations of AGM-2 and AGM-3 are reported on the right panel and are in line with their respective performance outlined in Table 1.

The AGM-3 proposal density declines to zero in π_1 's low density regions, an apparently appealing feature. However, AGM shrinks its proposal density in locations that have not been visited by the chain, which can be problematic if the sample space is large and takes some time to be reasonably explored (see Section 5).

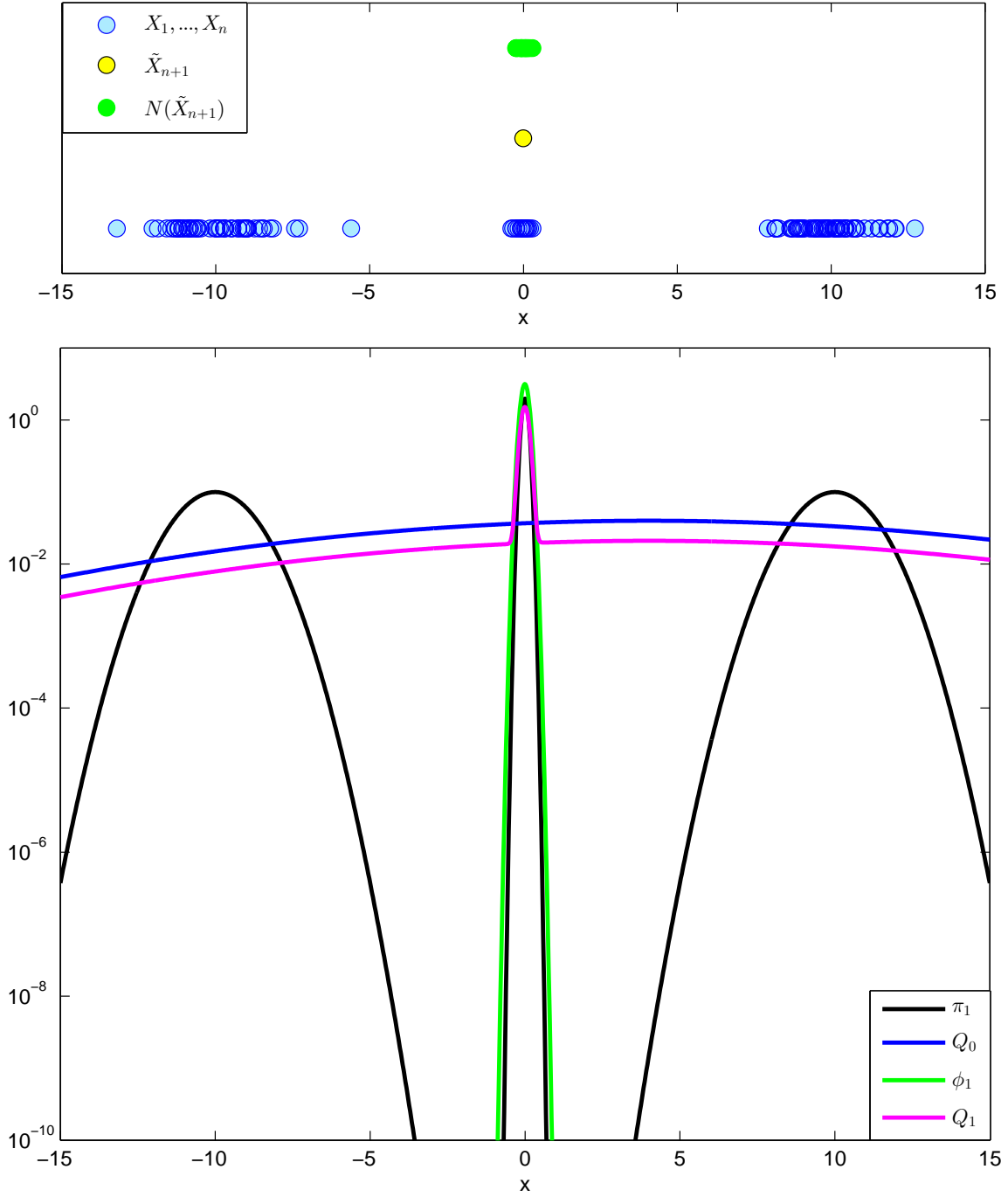


Figure 1: Illustration of one AIMM increment for the target π_1 . Top: states X_1, \dots, X_n of the Markov chain, proposed new state \tilde{X}_{n+1} activating the increment process (*i.e* satisfying $W_n(\tilde{X}_{n+1}) \geq W^*$) and neighborhood of \tilde{X}_{n+1} . Bottom: target π_1 , defensive kernel Q_0 , first increment ϕ_1 and updated kernel Q_1 plotted in log-lin scale.

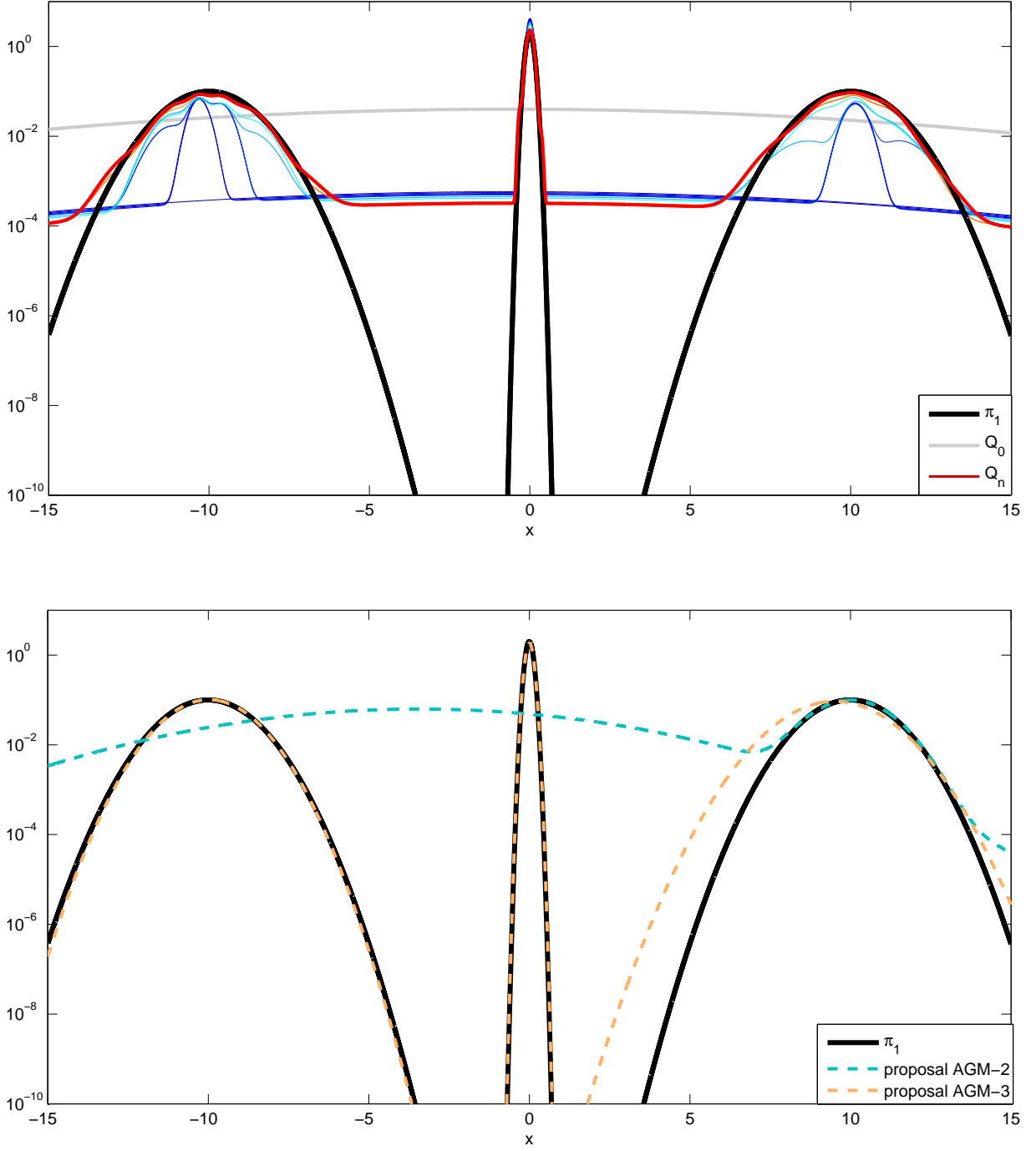


Figure 2: Illustration of AIMM and AGM sampling from the target π_1 for $n = 20,000$ iterations. Top: sequence of proposals from Q_0 to Q_n produced by AIMM. Bottom: proposals produced by AGM-2 and AGM-3. Both plots are in log-lin scale.

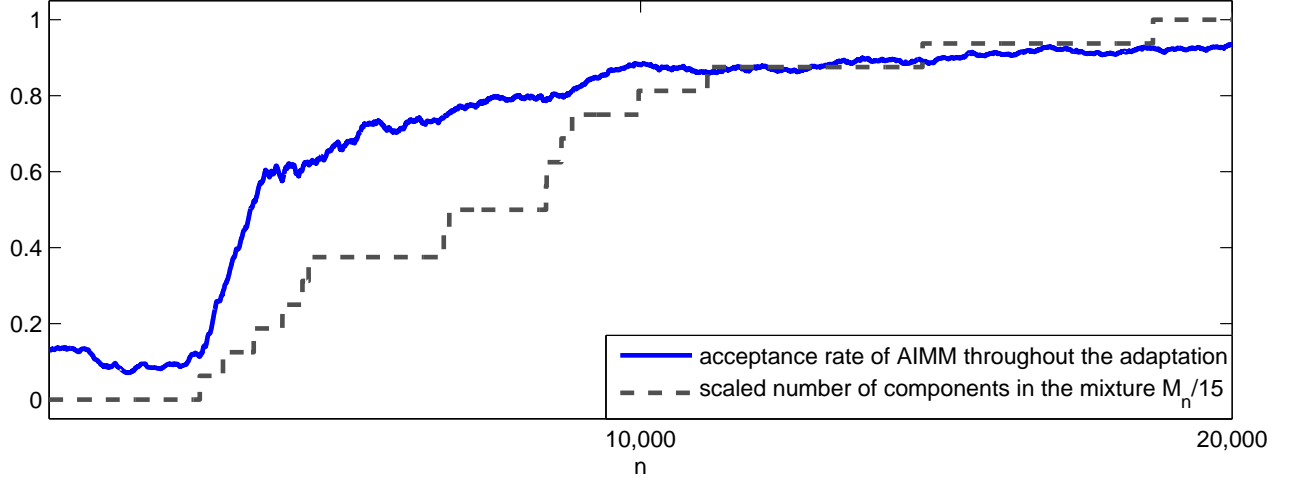


Figure 3: Effect of AIMM increment's on the sampling efficiency for the target π_1 - acceptance rate and number of increments throughout $n = 20,000$ iterations of AIMM.

4 Theoretical study

4.1 Notation

Let ν and μ be two probability measures defined on $(\mathbf{X}, \mathcal{X})$. Recall that the Kullback–Leibler (KL) divergence between ν and μ is defined as:

$$\text{KL}(\nu, \mu) = \int_{\mathbf{X}} \log \frac{\nu(x)}{\mu(x)} d\nu(x).$$

Moreover, the total variation distance between ν and μ can be written as

$$\|\nu - \mu\| = \sup_{A \in \mathcal{X}} |\nu(A) - \mu(A)| = \frac{1}{2} \int_{\mathbf{X}} |\mu(x) - \nu(x)| d\rho(x),$$

where the latter equality holds if μ and ν are both dominated by a common measure ρ . Note also that Pinsker's inequality allows one to bound the total variation in terms of KL divergence:

$$\|\nu - \mu\| \leq \sqrt{\frac{1}{2} \text{KL}(\nu, \mu)}. \quad (6)$$

Finally, let $\mathcal{F}_n = \sigma(X_1, \dots, X_n, \phi_1, \dots, \phi_{M_n})$ be the sigma-algebra generated by the sequence of random variables created by AIMM, and let \mathbb{P}_n be the probability distribution induced by AIMM given \mathcal{F}_n .

4.2 Adaptation of the incremental proposal kernel

Let $\tilde{Q}_n = \sum_{\ell=1}^{M_n} \beta_\ell \phi_\ell / \bar{\beta}_n$, with $\bar{\beta}_n = \sum_{\ell=1}^{M_n} \beta_\ell$, be the incremental part of the proposal kernel and the sequence of functions $\tilde{\text{KL}}_n : \mathcal{X} \rightarrow \mathbb{R}^+$ defined for all $A \in \mathcal{X}$ by

$$\tilde{\text{KL}}_n(A) = \int_A \log \frac{\pi(x)}{\tilde{Q}_n(x)} d\pi(x).$$

For all $n \in \mathbb{N}$, we will write $\tilde{\text{KL}}_n$ for $\tilde{\text{KL}}_n(\mathbf{X})$ as this coincides with the KL divergence between π and \tilde{Q}_n . The following proposition shows that as the adaptation progresses, a new increment is increasingly likely to yield a reduction in the KL divergence between π and \tilde{Q}_n . The proof is given in Appendix A.

Proposition 1. *Consider the sequence $\{\tilde{D}_n\}_{n>0}$, defined as the increments of the consecutive KL divergences, i.e. $\tilde{D}_n = \tilde{\text{KL}}_{n+1} - \tilde{\text{KL}}_n$. Then:*

- (i) $\mathbb{P}_n(\tilde{D}_n \leq 0) \geq u_n$ where $\{u_n\}_{n \in \mathbb{N}}$ is a non-decreasing sequence.
- (ii) \tilde{D}_n converges to zero in probability.

From Proposition 1, we derive the two following corollaries. Corollary 1 extends Proposition 1 to AIMM's proposal kernel Q_n while Corollary 2 states that the sequence of proposals will converge, in the sense of their KL divergence, even though it might increment infinitely often. Proofs of both corollaries are presented in Appendices B and C.

Corollary 1. *Let $\{D_n\}_{n>0}$ be the sequence of KL divergences between the target distribution and the independence proposal kernel Q_n (3), that is, $D_n = \text{KL}_{n+1} - \text{KL}_n$, where $\text{KL}_n = \text{KL}(\pi, Q_n)$. Then D_n converges to zero in probability.*

Corollary 2. *The sequence of random variables $\{\text{KL}(Q_n, Q_{n+1})\}_{n>0}$ goes to zero in probability.*

4.3 Ergodicity of AIMM

The ergodicity of adaptive MCMC methods is classically established by proving that the chain satisfies the *Diminishing adaptation* and *Containment* conditions of Roberts and Rosenthal (2007).

Condition 1. *Diminishing adaptation.*

The stochastic process $\{\Delta_n, n \in \mathbb{N}\}$, defined as

$$\Delta_n = \sup_{x \in \mathbf{X}} \|K_{n+1}(x, \cdot) - K_n(x, \cdot)\|,$$

converges to 0 in probability.

Condition 2. *Containment.*

For all $\epsilon > 0$, there exists $N \in \mathbb{N}$ such that

$$\forall x \in \mathbf{X}, \quad \|K_n(x, \cdot) - \pi\|^N < \epsilon.$$

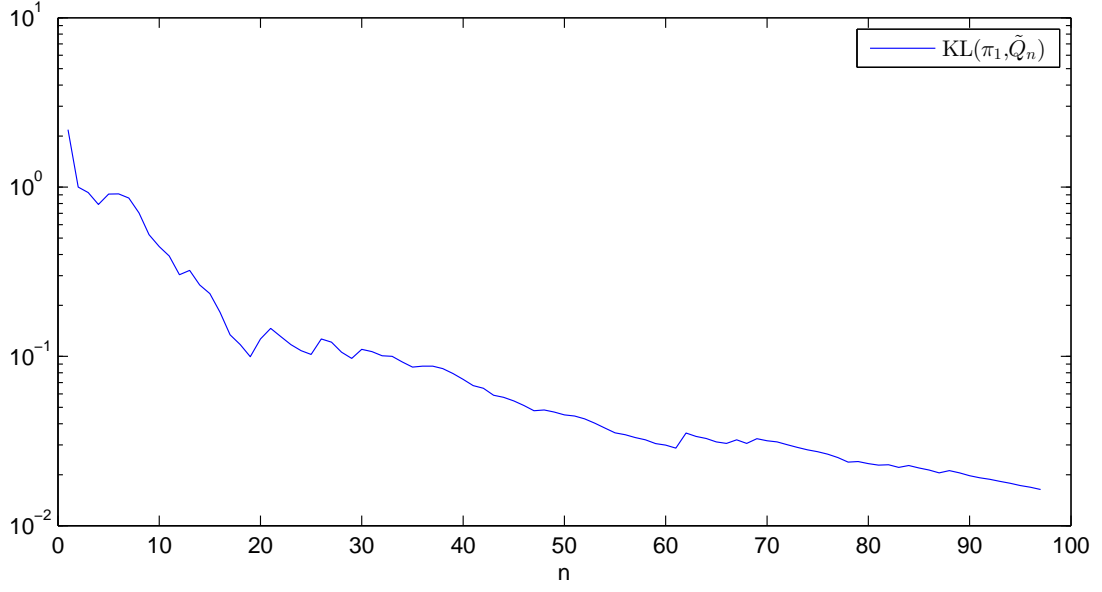
Proposition 2. Assume that the weight function W_n is π -almost everywhere bounded. Then, the adaptive Markov chain produced by AIMM satisfies Conditions 1 and 2 above and is thus ergodic:

$$\lim_{n \rightarrow \infty} \|\mathbb{P}_n(X_n \in \cdot) - \pi\| = 0.$$

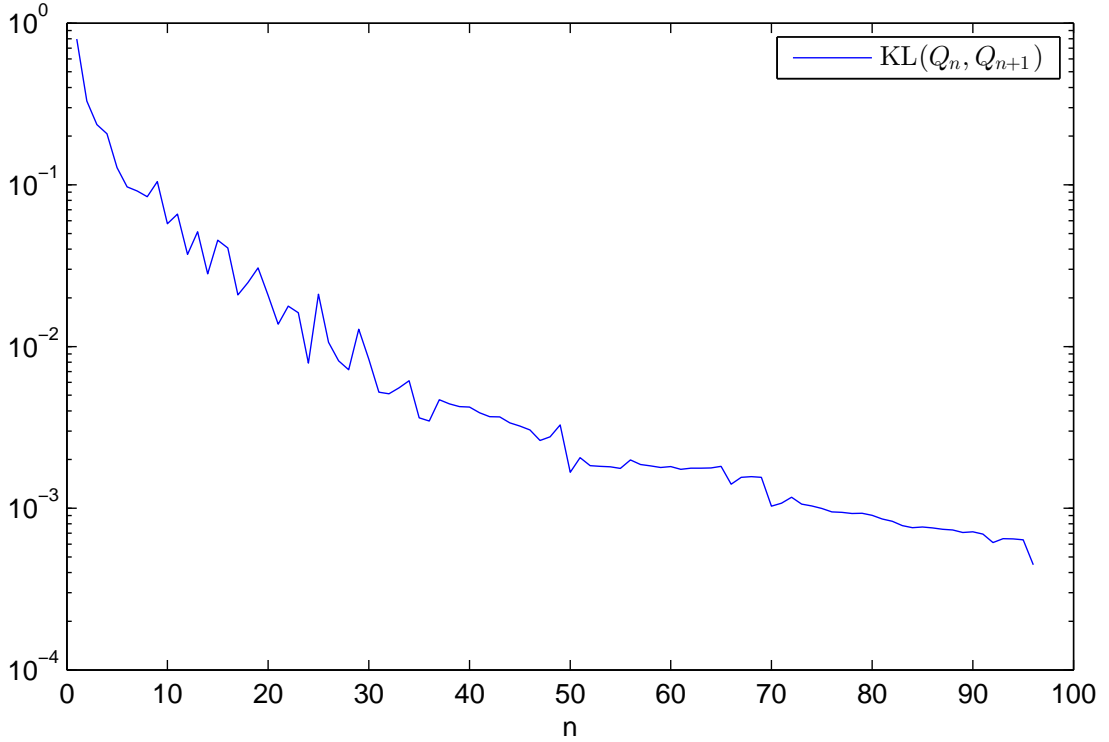
The proof of this proposition is outlined in Appendix D.

4.4 Example 1 (ctd.)

We illustrate the previous theoretical results in the context of Example 1. Figure 4 reports a realization of the two random processes $\{\text{KL}(\pi_1, \tilde{Q}_n)\}_n$ and $\{\text{KL}(Q_n, Q_{n+1})\}_n$ introduced in the two previous subsections. Here, AIMM is implemented to target π_1 , in a long run of 200,000 iterations. In these plots, and with some abuse of notation, n refers to the index of increments M_n and not to the actual iterations of AIMM. More precisely, Figure 4a empirically verifies Proposition 1, showing that as the adaptation progresses it becomes more unlikely that an increment increases $\text{KL}(\pi_1, \tilde{Q}_n)$ (Prop. 1 (i)) and that the KL divergence between π_1 and \tilde{Q}_n decreases and stabilizes (Prop. 2 (ii)). Figure 4b is in agreement with Corollary 2 and shows that the discrepancy between two consecutive proposals vanishes as AIMM increments the proposal.



(a) Evolution of the overall decrease in KL divergence between π_1 and the incremental part of Q_n , plotted in the log-lin scale.



(b) Evolution of the overall decrease in KL divergence between two consecutive proposals Q_n and Q_{n+1} , plotted in the log-lin scale.

Figure 4: Empirical illustration of Proposition 1 (top) and Corollary 2 (bottom) on the target π_1 .

5 Simulations

In this Section, we consider three target distributions:

- π_2 , the banana shape distribution used in Haario et al. (2001);
- π_3 , the ridge like distribution used in Raftery and Bao (2010);
- π_4 , the bimodal distribution used in Raftery and Bao (2010).

Each of these distributions has a specific feature, resulting in a diverse set of challenging targets. π_2 is heavy tailed, π_3 has a narrow and ridge-like support and π_4 has two distant modes. In order to emulate the setup of a real data example, we consider the generic distributions π_2 and π_4 in different dimensions: $d \in \{2, 10, 40\}$ for π_2 and $d \in \{4, 10\}$ for π_4 . We compare AIMM with several other algorithms that are briefly described in Appendix E.1 using some performance indicators that are defined, explained and justified in Appendix E.2. We used the following default parameters for AIMM:

$$W^* = d, \quad \gamma = 0.5, \quad \tau = 0.5, \quad \kappa = 0.1, \quad N_0 = 1,000\sqrt{d}.$$

For each scenario, we implemented AIMM with different thresholds W^* valued around d to monitor the tradeoff between adaptation and computational efficiency of the algorithm. In our experience, the choice $W^* = d$ has worked reasonably well in a wide range of examples. As the dimension of the state space increases a higher threshold is required, too many kernels being created otherwise. However, a satisfactory choice of W^* may vary depending on the target distribution and the computational budget.

5.1 Banana shape target distribution

Example 2. Let $\mathbf{X} = \mathbb{R}^d$ and $\pi_2(x) = \Psi_d(f_b(x); m, S)$ where $\Psi_d(\cdot; m, S)$ is the d -dimensional Gaussian density with mean m and covariance matrix S , and $f_b : \mathbb{R}^d \rightarrow \mathbb{R}^d$ is the mapping

defined, for any $b \in \mathbb{R}$, by

$$f_b : \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_d \end{pmatrix} \rightarrow \begin{pmatrix} x_1 \\ x_2 + b x_1^2 - 100b \\ x_3 \\ \vdots \\ x_d \end{pmatrix}.$$

We consider π_2 in dimensions $d = 2$, $d = 10$ and $d = 40$ and refer to the marginal of π_2 in the i -th dimension as $\pi_2^{(i)}$. The parameters $m = \mathbf{0}_d$ and $S = \text{diag}([100, \mathbf{1}_{d-1}])$ are held constant. The target is banana-shaped along the first two dimensions, and is challenging since the high density region has a thin and wide ridge-like shape with narrow but heavy tails. Unless otherwise stated, we use $b = 0.1$ which accentuates these challenging features of π_2 . We first use the banana shape target π_2 in dimension $d = 2$ to study the influence of AIMM's user-defined parameters on the sampling efficiency.

5.1.1 Influence of the defensive distribution

With limited prior knowledge of π , choosing Q_0 can be challenging. Here we consider three defensive distributions Q_0 , represented by the black dashed contours in Figure 5. The first two are Gaussian, respectively, located close to the mode and in a nearly zero probability area and the last one is the uniform distribution on the set $\mathfrak{S} = \{x \in \mathbf{X}, x_1 \in (-50, 50), x_2 \in (-100, 20)\}$. Table 2 and Figure 5 show that AIMM is robust with respect to the choice of Q_0 . Even when Q_0 yields a significant mismatch with π_2 (second case), the incremental mixture reaches high density areas, progressively uncovering higher density regions. The price to pay for a poorly chosen Q_0 is that more components are needed in the incremental mixture to match π_2 , as shown by Table 2. The other statistics reported in Table 2 are reasonably similar for the three choices of Q_0 .

5.1.2 Influence of the threshold

The threshold W^* controls the number of kernels M_n created by AIMM, and hence its computational efficiency. We consider the same setup as in the previous subsection with

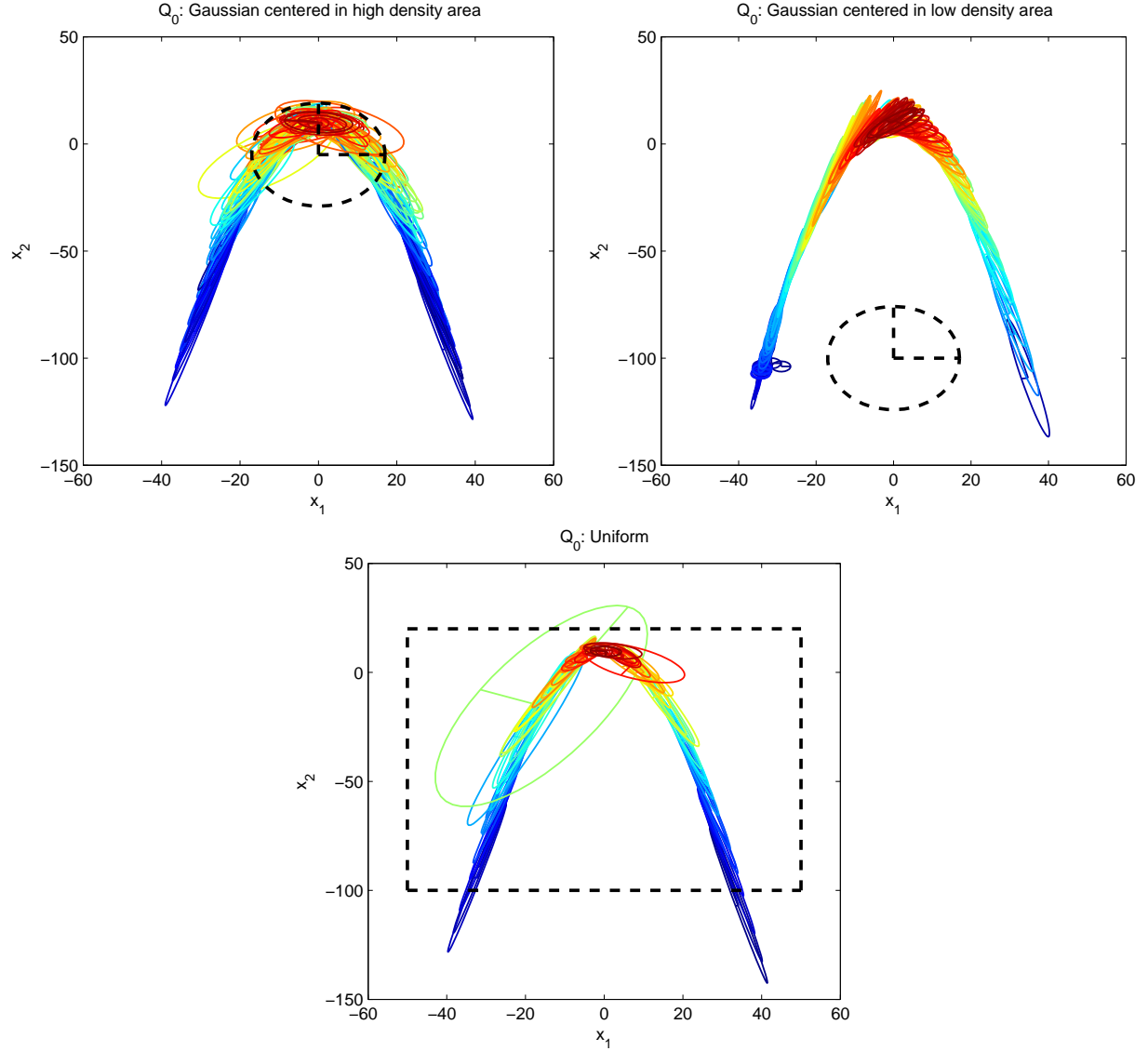


Figure 5: π_2 target, $d = 2$ - Incremental mixture created by AIMM for three different initial proposals. Top row: Q_0 is a Gaussian density (the region inside the black dashed ellipses contains 75% of the Gaussian mass) centred on a high density region (left) and a low density region (right). Bottom row: Q_0 is Uniform (with support corresponding to the black dashed rectangle). The components $\phi_1, \dots, \phi_{M_n}$ of the incremental mixture obtained after $n = 100,000$ MCMC iterations are represented through (the region inside the ellipses contains 75% of each Gaussian mass). The color of each ellipse illustrates the corresponding component's relative weight β_ℓ (from dark blue for lower weights to red).

Table 2: π_2 target, $d = 2$ - Influence of the initial proposal Q_0 on AIMM outcome after $n = 100,000$ MCMC iterations (replicated 20 times). M_n is the number of components created by AIMM, ESS is the effective sample size, ACC is the acceptance rate, KL is the KL divergence between π_2 and the Markov chain distribution, JMP is the average distance between two consecutive states of the Markov chain and EFF is a time-normalized ESS.

Q_0	M_n	ESS	ACC	KL	JMP	EFF ($\times 10^{-4}$)
Gaussian on a high density region	41	.19	.35	.59	197	11
Gaussian on a low density region	157	.23	.37	.71	245	4
Uniform on \mathfrak{S}	39	.24	.38	.62	320	11

the uniform defensive distribution Q_0 and report the outcome of the AIMM algorithm in Table 3a. As expected, the lower the threshold W^* , the larger the number of kernels and the better the mixing. The adaptive Markov kernel stabilizes, as illustrated by Figure 6a, resulting from the fact that the event $\mathcal{E}_n = \{W_n(\tilde{X}_{n+1}) > W^*, \tilde{X}_{n+1} \sim Q_n\}$ occurs less often as n increases. Figure 6b agrees with the theoretical result of Section 4, as it shows that the event $\{\text{KL}(\pi, Q_n) > \text{KL}(\pi, Q_{n+1})\}$ holds with a probability that increases with n , implying that the increment process improves the proposal distribution with high probability.

As W^* decreases, the KL divergence between π_2 and the chain reduces while the CPU cost increases since more components are created. Therefore, the sampling efficiency is best for an intermediate threshold, taken here as $\log W^* = 1$. Finally, when the threshold is too low, the distribution of the chain converges more slowly to π_2 ; see e.g. Table 3a where KL (defined as the K–L divergence between π and the sample path of the Markov chain; see Section E.2) is larger for $\log W^* = 0.25$ than for $\log W^* = 0.5$. Indeed when $W^* \ll 1$, too many kernels are created to support high density areas and this slows down the exploration process of the lower density regions.

5.1.3 Speeding up AIMM

The computational efficiency of AIMM depends, to a large extent, on the number of components added in the proposal distribution, M_n . For this reason we consider a slight modifi-

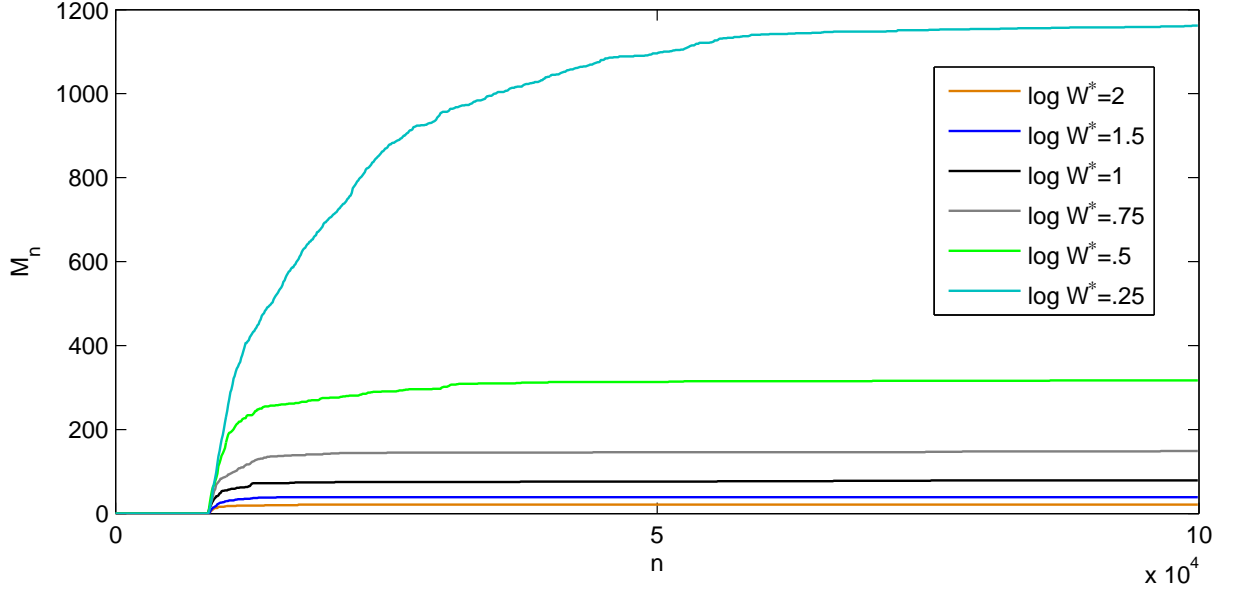
Table 3: π_2 target, $d = 2$ - Influence of the threshold W^* on AIMM and f-AIMM outcomes after $n = 100,000$ MCMC iterations (replicated 20 times).

(a) AIMM

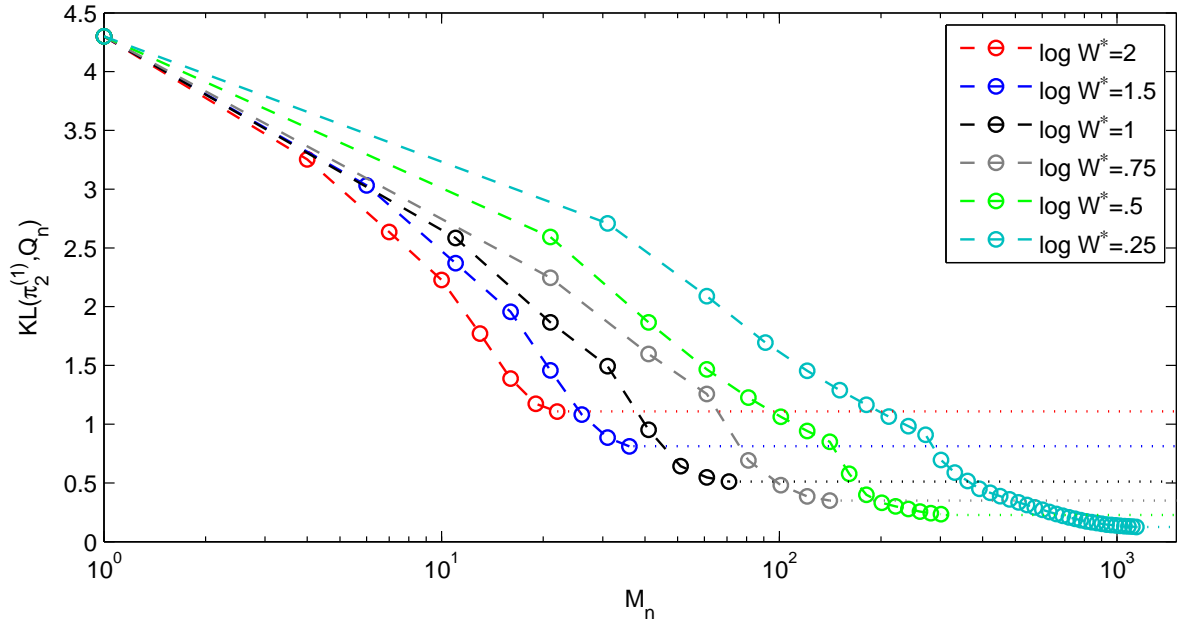
$\log W^*$	M_n	ESS	ACC	KL	JMP	CPU	EFF ($\times 10^{-4}$)
10	0	.03	.01	14.06	10	45	2
2	21	.16	.27	.72	169	154	10
1.5	39	.24	.38	.62	235	245	10
1	79	.37	.53	.54	331	330	12
.75	149	.48	.64	.53	286	658	7
.5	317	.64	.75	.51	451	2,199	3
.25	1162	.71	.87	.54	505	6,201	1

(b) f-AIMM

$\log W^*$	M_{\max}	ESS	ACC	KL	JMP	CPU	EFF ($\times 10^{-4}$)
1.5	25	.29	.51	.59	268	255	11
.75	100	.53	.69	.53	409	421	13
.5	200	.67	.80	.51	474	1,151	6



(a) Evolution of the number of kernels M_n created by AIMM for different thresholds.



(b) Evolution of the KL divergence between π_2 and the incremental proposal Q_n , plotted in lin-log scale.

Figure 6: π_2 target, $d = 2$ - AIMM's incremental mixture design after $n = 100,000$ MCMC iterations.

cation of the original AIMM algorithm to limit the number of possible components, thereby improving the computational speed of the algorithm. This variant of the algorithm is outlined below and will be referred to as *fast* AIMM and denoted by f-AIMM:

- Let M_{\max} be the maximal number of components allowed in the incremental mixture proposal. If $M_n > M_{\max}$, only the last M_{\max} added components are retained, in a moving window fashion. This *truncation* has two main advantages, (i) approximately linearizing the computational burden once M_{\max} is reached, and (ii) forgetting the first, often transient components used to jump to high density areas (see *e.g* the loose ellipses in Figure 5, especially the very visible ones in the bottom panel).
- The threshold W^* is adapted at the start of the algorithm in order to get the incremental process started. If the initial proposal Q_0 misses π , then $\mathbb{E}_0\{W_0(\tilde{X})\} \ll 1$ and the initial threshold W_0^* should be set as low as possible, to start the adaptation. However, as soon as the proposal is incremented, $\mathbb{E}_n\{W_n(\tilde{X})\}$ increases and leaving the threshold at its initial level will result in $\mathbb{P}_n\{W_n(\tilde{X}) > W^*\} \approx 1$, i.e. in adding too many irrelevant components. The threshold adaptation produces a sequence of thresholds $\{W_n^*\}_n$ such that $Q_n\{W_n(X) > W_n^*\} \approx 10^{-3}$. Since sampling from Q_n (a mixture of Gaussian distributions), can be performed routinely, W_n^* is derived from a Monte Carlo estimation. As no precision is required, the Monte Carlo approximation should be rough in order to limit the computational burden generated by the threshold adaptation. Also, those samples used to set W_n^* can be recycled and serve as proposal states for the Markov chain so that this automated threshold mechanism comes for free. The threshold adaptation stops when the adapted threshold W_n^* reaches a neighborhood of the prescribed one W^* , here defined as $|W_n^* - W^*| < 1$.

Table 3 shows that f-AIMM outperforms the original AIMM, with some setups being nearly twice as efficient; compare, *e.g.* AIMM and f-AIMM with $\log W^* = .5$ in terms of efficiency (last column). Beyond efficiency, comparing KL for a given number of mixture components (M_n), shows that π_2 is more quickly explored by f-AIMM than AIMM.

5.1.4 Comparison with other samplers

We now compare f-AIMM with four other samplers of interest: the adaptive algorithms AMH and AGM and their non-adaptive counterparts, random walk Metropolis-Hastings (RWMH) and Independent Metropolis (IM). We consider π_2 in dimensions $d = 2, 10, 40$. Table 4 summarizes the simulation results and emphasizes the need to compare the asymptotic variance related statistics (ESS, ACC, JMP) jointly with the convergence related statistic (KL). Indeed, from the target in dimension $d = 2$, one can see that AGM yields the best ESS but is very slow to converge to π_2 as it still misses a significant portion of the state space after $n = 200,000$ iterations; see the KL column in Table 4 and Appendix E.3 (especially Figure 14) where we shed light on AGM’s performance in this situation.

AIMM seems to yield the best tradeoff between convergence and variance in all dimensions. Inevitably, as d increases, ESS shrinks and KL increases for all algorithms but AIMM still maintains a competitive advantage over the other approaches. It is worth noting that, if computation is not an issue, AIMM is able to reach a very large ESS (almost one), while having converged to the target. In such a situation, AIMM is essentially simulating *i.i.d.* draws from π_2 .

Figure 7 compares the kernel density approximation of the Markov chain’s marginal distribution of the first component for the five algorithms, with the true marginal $\pi_2^{(1)} = \mathcal{N}(0, 10)$. AIMM converges to π_2 faster than the other samplers, which need much more than $n = 200,000$ iterations to converge. The bumpy shape of the AGM and IM marginal densities is caused by the accumulation of samples in some locations and reflects the slow convergence of these algorithms.

The other four algorithms struggle to visit the tail of the distribution, see Table 5. Thus AIMM is doing better than (i) the non-independent samplers (RWMH and AMH) that manage to explore the tail of a target distribution but need a large number of transitions to return there, and (ii) the independence samplers (IM and AGM) that can quickly explore different areas but fail to venture into the lower density regions. An animation corresponding to the exploration of π_2 in dimension 2 by AIMM, AGM and AMH can be found at <http://mathsci.ucd.ie/~fmaire/AIMM/banana.html>.

Table 4: π_2 target, $d \in \{2, 10, 40\}$ - Comparison of f-AIMM with RWMH, AMH, AGM and IM. Statistics were estimated from 20 replications of the five corresponding Markov chains after $n = 200,000$ iterations. AGM fails to give sensible results for $d = 40$. †: the ESS column are $\times 10^{-3}$; ‡: the EFF column are $\times 10^{-6}$.

(a) π_2 in dimension $d = 2$

	$\log W^*$	M_{\max}	ESS †	ACC	KL	JMP	CPU	EFF ‡
f-AIMM	1.5	25	290	.51	.59	268	355	810
f-AIMM	.5	200	670	.80	.51	474	1,751	380
RWMH	—	—	1	.30	3.75	2	48	17
AMH	—	—	4	.23	2.69	7	60	63
AGM	—	100	800	.78	29.2	155	2,670	230
IM	—	—	6	.01	10.1	8	92	63

(b) π_2 in dimension $d = 10$

	$\log W^*$	M_{\max}	ESS †	ACC	KL	JMP	CPU	EFF ‡
f-AIMM	3	50	110	.27	1.46	135	1,160	94
f-AIMM	2.5	150	170	.37	.84	205	1,638	100
RWMH	—	—	1	.22	9.63	1.1	61	15
AMH	—	—	1.4	.17	5.29	2.5	143	10
AGM	—	100	600	.67	119	153	4,909	120
IM	—	—	.7	10^{-3}	97	.9	316	2

(c) π_2 in dimension $d = 40$

	$\log W^*$	M_{\max}	ESS †	ACC	KL	JMP	CPU	EFF ‡
f-AIMM	5	200	4	.04	12.1	24	4,250	.7
f-AIMM	4	1,000	17	.08	5.9	61	10,012	1.8
RWMH	—	—	.8	.24	11.6	.8	71	11
AMH	—	—	.8	.13	6.7	1.5	1,534	.7
IM	—	—	.5	10^{-4}	147.1	.03	1,270	.6

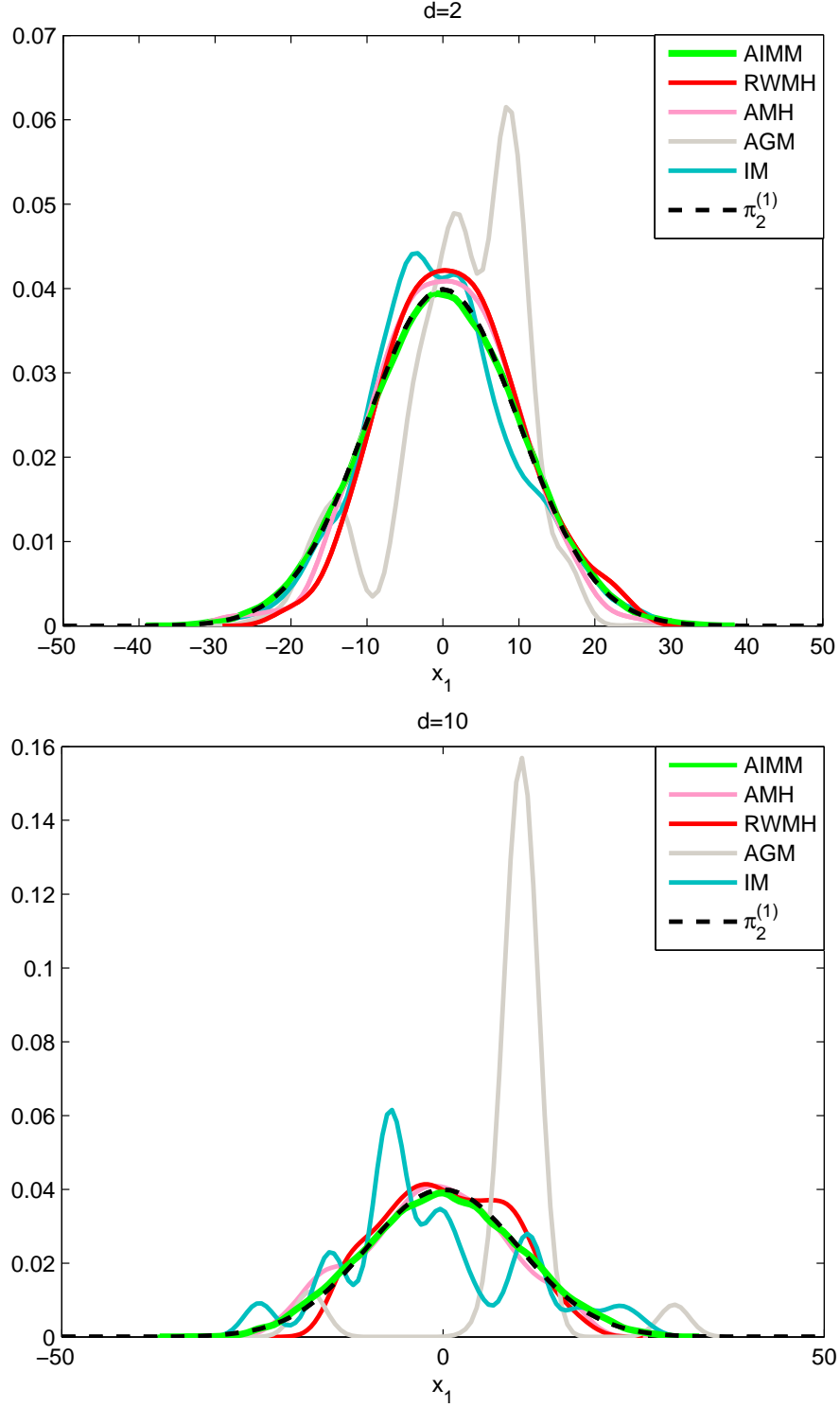


Figure 7: π_2 target, $d \in \{2, 10\}$ - Kernel approximation of the marginal density $\pi_2^{(1)}$ provided by $n = 200,000$ samples of the five Markov chains AIMM, AMH, RWMH, AGM and IM for the banana shape target example in dimensions $d = 2$ (top) and $d = 10$ (bottom).

Table 5: π_2 target, $d \in \{2, 10, 40\}$ - Tail exploration achieved by the five different samplers. Tail events are defined as $\{\text{TAI}_1\} = \{X^{(2)} < -28.6\}$ and $\{\text{TAI}_2\} = \{X^{(2)} < -68.5\}$ so that $\pi_2\{\text{TAI}_1\} = 0.05$ and $\pi_2\{\text{TAI}_2\} = 0.005$. $\{\text{RET}_1\}$ and $\{\text{RET}_2\}$ corresponds to the expected returning time to the tail events $\{\text{TAI}_1\}$ and $\{\text{TAI}_2\}$, respectively. Estimation of these statistics was achieved with $n = 200,000$ Markov transitions of the corresponding samplers, repeated 20 times.

		M_{\max}	TAI ₁	RET ₁	TAI ₂	RET ₂
$d = 2$	f-AIMM	25	.05	43	.003	557
	f-AIMM	200	.05	23	.005	281
	RWMH	–	.04	2,945	.002	69,730
	AMH	–	.04	3,676	.004	74,292
	AGM	100	.01	1,023	$5.0 \cdot 10^{-5}$	155,230
	IM	–	.05	1,148	.004	8,885
$d = 10$	f-AIMM	50	.04	98	.001	5,463
	f-AIMM	150	.05	57	.002	1,708
	RWMH	–	.03	30,004	0	∞
	AMH	–	.04	4,081	.001	166,103
	AGM	100	.04	120,022	.001	180,119
	IM	–	.05	13,140	.003	103,454
$d = 40$	f-AIMM	200	.01	1,350	0	∞
	f-AIMM	1000	.05	175	$2.0 \cdot 10^{-4}$	16,230
	RWMH	–	.002	158,000	0	∞
	AMH	–	0	∞	0	∞
	IM	–	.006	28,552	0	∞

5.2 Ridge like example

Example 3. Let $\mathsf{X} = \mathbb{R}^6$ and $\pi_3(x) \propto \Psi_6(x; \mu_i, \Gamma_i) \Psi_4(g(x); \mu_o, \Gamma_o)$, where $\Psi_d(\cdot; m, S)$ is the d -dimensional Gaussian density with mean m and covariance matrix S . The target parameters $(\mu_i, \mu_o) \in \mathbb{R}^6 \times \mathbb{R}^4$ and $(\Gamma_i, \Gamma_o) \in \mathcal{M}_6(\mathbb{R}) \times \mathcal{M}_4(\mathbb{R})$ are, respectively, known means and covariance matrices and g is a nonlinear deterministic mapping $\mathbb{R}^6 \rightarrow \mathbb{R}^4$ defined as:

$$g(x_1, \dots, x_6) = \begin{cases} \prod_{i=1}^6 x_i, \\ x_2 x_4, \\ x_1/x_5, \\ x_3 x_6. \end{cases}$$

In this context, $\Psi_6(\cdot; \mu_i, \Gamma_i)$ can be regarded as a prior distribution and $\Psi_4(\cdot; \mu_o, \Gamma_o)$ as a likelihood, the observations being some functional of the hidden parameter $x \in \mathbb{R}^6$.

Such a target distribution often arises in physics. Similar target distributions often arise in Bayesian inference on deterministic mechanistic models. They are hard to sample from because the probability mass is concentrated around thin curved manifolds. We compare f-AIMM with RWMH, AMH, AGM and IM first in terms of their mixing properties; see Table 6. We also ensure that the different methods agree on the mass localisation by plotting a pairwise marginal; see Figure 8. In this example, f-AIMM clearly outperforms the four other methods in terms of both convergence and variance statistics. We observe in Figure 8, that f-AIMM is the only sampler able to discover a secondary mode in the marginal distribution of the second and fourth target components (X_2, X_4) .

5.3 Bimodal distribution

Example 4. In this last example, π_4 is a posterior distribution defined on the state space $\mathsf{X} = \mathbb{R}^d$, where $d \in \{4, 10\}$. The likelihood is a mixture of two d -dimensional Gaussian distributions with weight $\lambda = 0.5$, mean and covariance matrix as follows

$$\begin{aligned} \mu_1 &= \mathbf{0}_d, & \Sigma_1 &= AR_d(-.95), \\ \mu_2 &= \mathbf{1}_d, & \Sigma_2 &= AR_d(.95), \end{aligned}$$

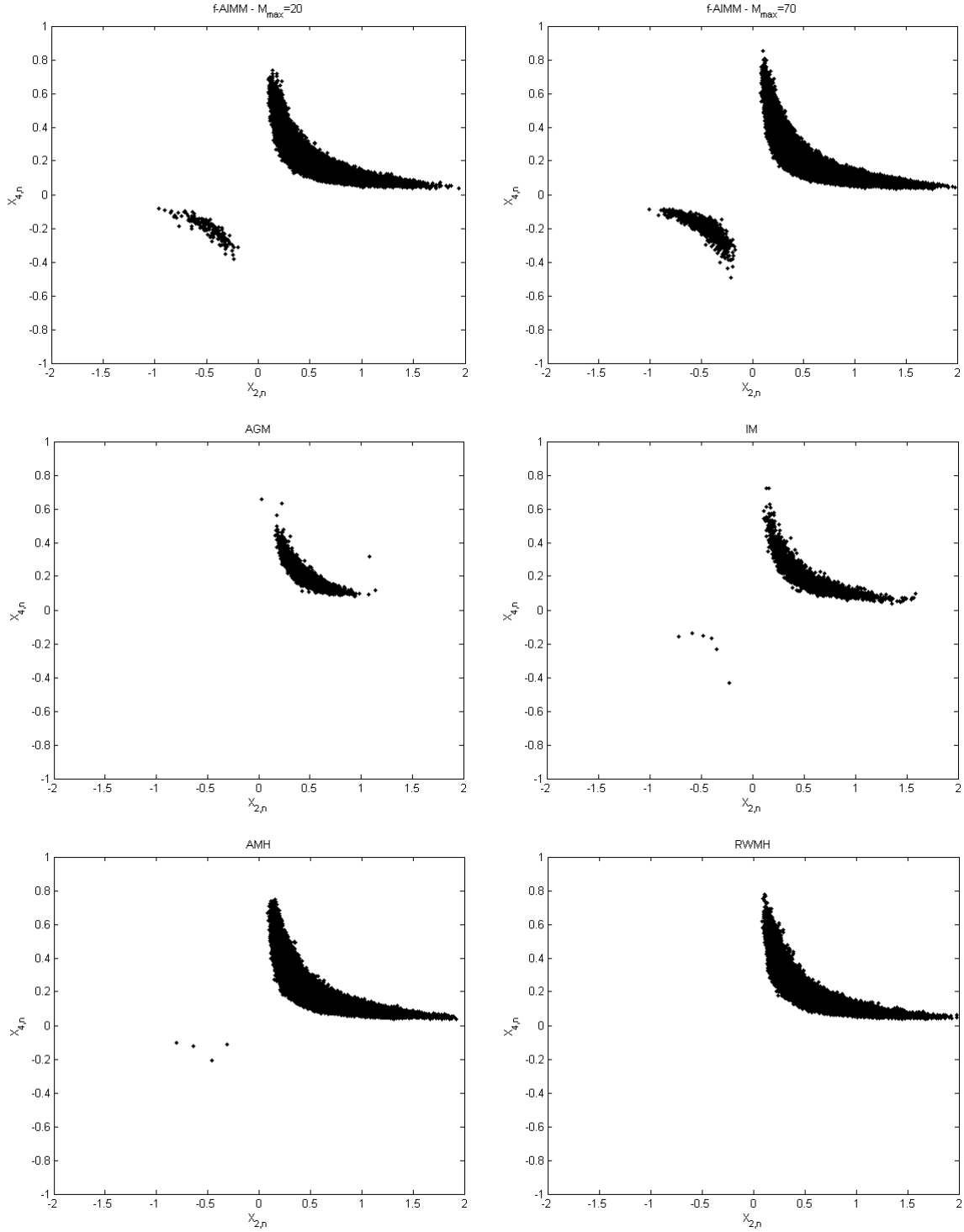


Figure 8: π_3 target, $d = 6$ - Samples $\{(X_{2,n}, X_{4,n}), n \leq 200,000\}$ from the five different algorithms: AIMM in two settings $M_{\max} = 20$ and $M_{\max} = 70$, AGM, IM, AMH and RWMH.

Table 6: π_3 target, $d = 6$ - Comparison of f-AIMM with the four other samplers after $n = 200,000$ iterations (replicated 10 times).

	W^*	M_{\max}	ACC	ESS	CPU	EFF	JMP
f-AIMM	10	20	.049	.015	274	$5.2 \cdot 10^{-5}$.08
f-AIMM	1	70	.169	.089	517	$1.7 \cdot 10^{-4}$.27
f-AIMM	.1	110	.251	.156	818	$1.9 \cdot 10^{-4}$.38
RWMH	–	–	.23	.001	109	$9.2 \cdot 10^{-6}$.007
AMH	–	–	.42	.002	2,105	$9.5 \cdot 10^{-7}$.004
AGM-MH	100	100	.009	.002	2,660	$3.4 \cdot 10^{-6}$.003
IM	–	–	.003	.001	199	$5.5 \cdot 10^{-6}$.003

where for all $\rho > 0$, $AR_d(\rho)$ is the d -dimensional first order autoregressive matrix whose coefficients are:

$$\text{for } 1 \leq (i, j) \leq d, \quad m_{i,j}(\rho) = \rho^{\max(i,j)-1}.$$

The prior is the uniform distribution $\mathcal{U}([-3, 12]^d)$. For f-AIMM and IM, Q_0 is set as this prior distribution, while for AGM, the centers of the Gaussian components in the initial proposal kernel are drawn according to the same uniform distribution. For AMH, the initial covariance matrix Σ_0 is set to be the identity matrix.

Figures 9 and 10 illustrate the experiment in dimensions $d = 4$ and $d = 10$. On the one hand, in both setups $M_{\max} = 100$ and $M_{\max} = 200$, f-AIMM is more efficient at discovering and jumping from one mode to the other. Allowing more kernels in the incremental mixture proposal will result in faster convergence; compare the settings with $M_{\max} = 100$ and $M_{\max} = 200$ in Figure 11. On the other hand, because of the distance between the two modes, RWMH visits only one while AMH visits both but in an unbalanced fashion. Figure 10 displays the samples from the joint marginal (X_1, X_2) obtained through f-AIMM (with two different values of M_{\max}), AMH and RWMH. Increasing the dimension from $d = 4$ to $d = 10$ makes AMH unable to visit the two modes after $n = 200,000$ iterations. As for AGM, the isolated samples reflect a failure to explore the state space. These facts are confirmed in Table 7 which highlights the better mixing efficiency of f-AIMM relative to the four other algorithms and in Table 8 which shows that f-AIMM is the only method which, after $n = 200,000$ iterations,

Table 7: π_4 target, $d \in \{4, 10\}$ - Comparison of f-AIMM with RWMH, AMH, AGM and IM after $n = 200,000$ iterations (replicated 100 times).

(a) π_4 in dimension $d = 4$

	W^*	M_{\max}	ACC	ESS	CPU	EFF	JMP
f-AIMM	5	100	.69	.30	408	$7.3 \cdot 10^{-4}$	68
RWMH	—	—	.23	$3.1 \cdot 10^{-3}$	93	$3.4 \cdot 10^{-5}$.09
AMH	—	—	.26	$2.7 \cdot 10^{-2}$	269	$1.0 \cdot 10^{-4}$.64
AGM-MH	100	100	$3.0 \cdot 10^{-3}$	$5.0 \cdot 10^{-4}$	3,517	$1.4 \cdot 10^{-7}$.29
IM	—	—	$3.3 \cdot 10^{-4}$	$5.9 \cdot 10^{-4}$	81	$7.3 \cdot 10^{-6}$.05

(b) π_4 in dimension $d = 10$

	W^*	M_{\max}	ACC	ESS	CPU	EFF	JMP
f-AIMM	20	100	.45	.18	1,232	$1.4 \cdot 10^{-4}$	116.4
f-AIMM	10	200	.64	.25	1,550	$1.6 \cdot 10^{-4}$	200.1
RWMH	—	—	.38	$7.2 \cdot 10^{-4}$	476	$1.7 \cdot 10^{-6}$.07
AMH	—	—	.18	$1.3 \cdot 10^{-3}$	2,601	$5.0 \cdot 10^{-7}$.57
AGM-MH	100	100	$2.6 \cdot 10^{-4}$	$5.0 \cdot 10^{-4}$	4,459	$1.1 \cdot 10^{-7}$	$7.1 \cdot 10^{-3}$
IM	—	—	$6.4 \cdot 10^{-5}$	$6.1 \cdot 10^{-4}$	473	$1.2 \cdot 10^{-6}$	$4.1 \cdot 10^{-3}$

visits the two modes with the correct proportion.

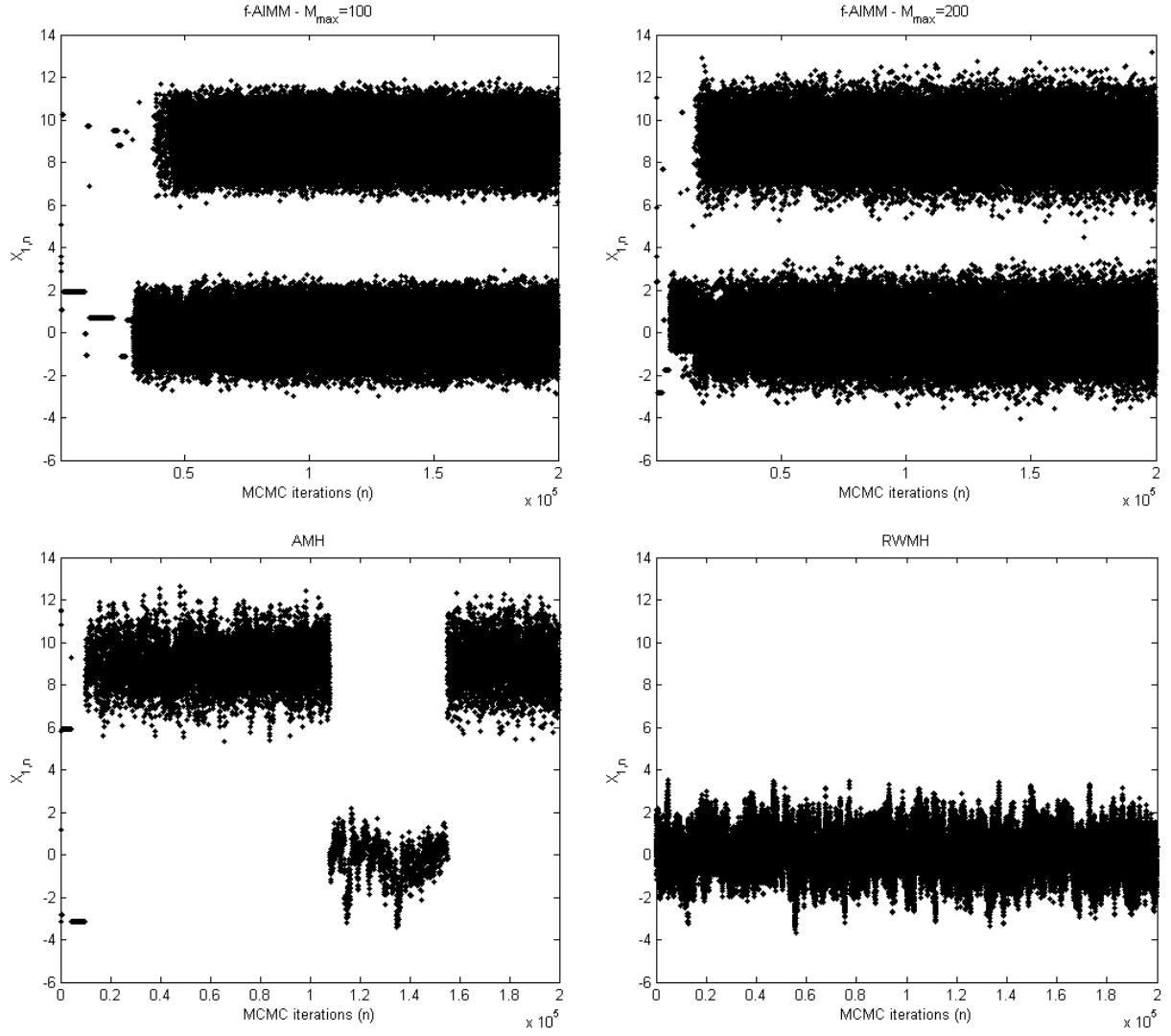


Figure 9: π_4 target, $d = 4$ - Samples $\{X_{1,n}, n \leq 200,000\}$ from f-AIMM in two settings $M_{\max} = 100$ and $M_{\max} = 200$, AMH and RWMH.

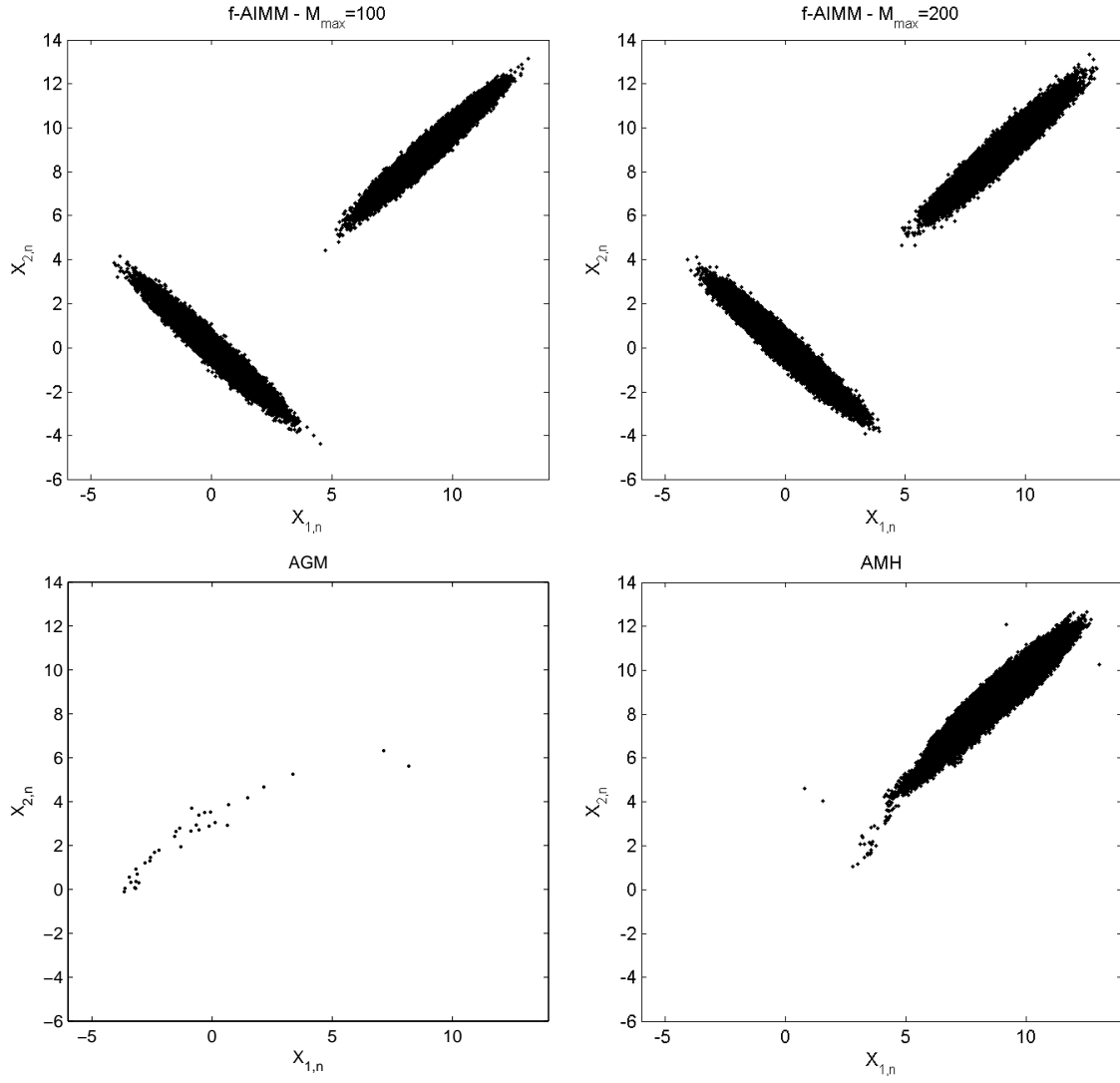


Figure 10: π_4 target, $d = 10$ - Samples $\{(X_{1,n}, X_{2,n}), n \leq 200,000\}$ from f-AIMM in two settings $M_{\max} = 100$ and $M_{\max} = 200$, AGM and AMH.

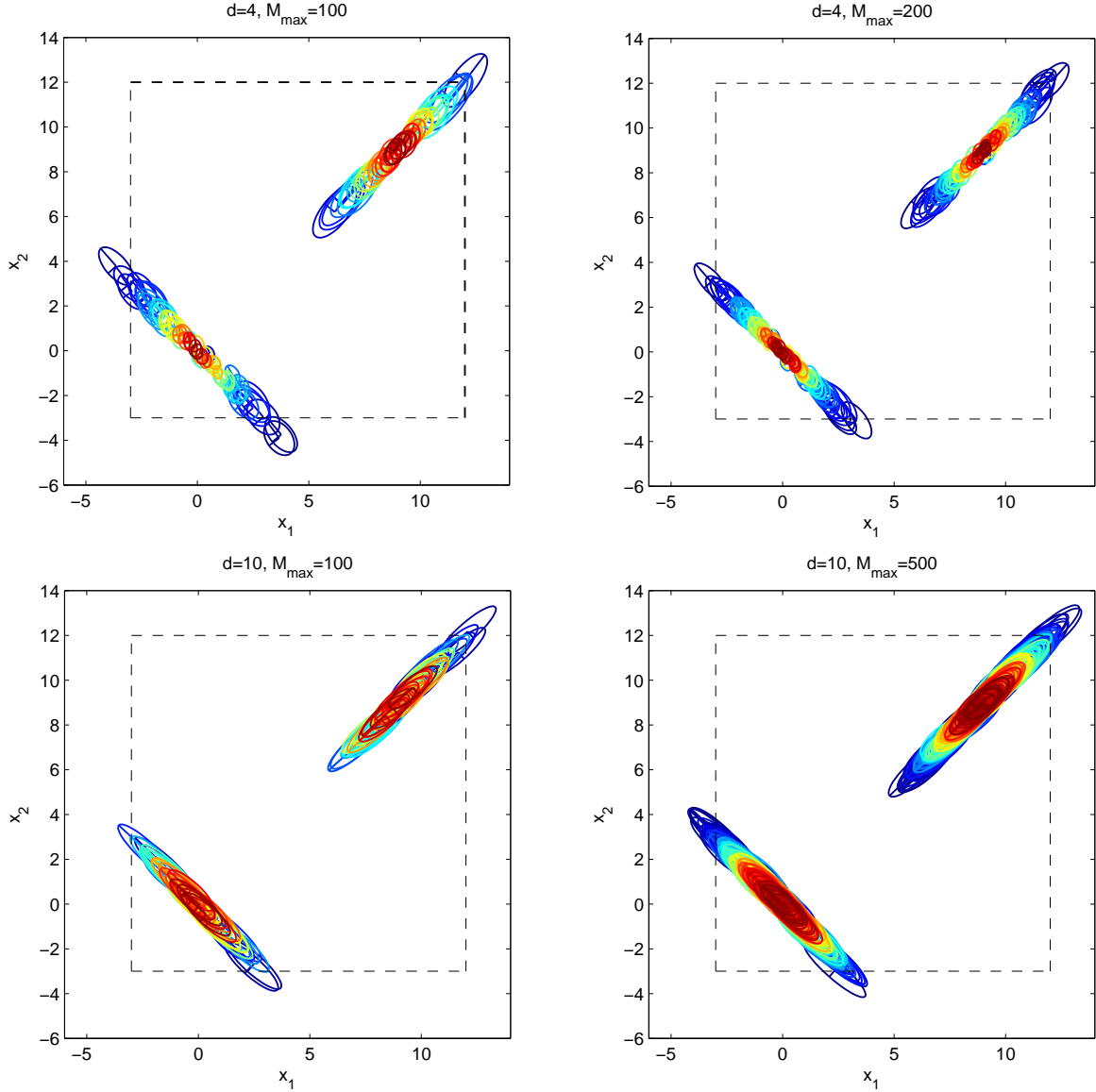


Figure 11: π_4 target, $d \in \{4, 10\}$ - Incremental mixture (colored ellipses) and uniform initial proposal (black dashed rectangle) designed by f-AIMM in the bimodal target π_4 , in dimensions $d = 4$ (top row) and $d = 10$ (bottom row). Distributions are projected onto the joint marginal space (X_1, X_2) . The colors stand for the weights $\{\beta_\ell\}_{\ell=1}^{M_n}$ attached to each component. The region inside the ellipses contains 75% of each Gaussian mass.

Table 8: π_4 target, $d \in \{4, 10\}$ - Mean Square Error (MSE) of the mixture parameter λ , for the different algorithms (replicated 100 times).

	W^*	M_{\max}	MSE(λ), $d = 4$	MSE(λ), $d = 10$
f-AIMM	5	100	.0001	.06
f-AIMM	20	100	.0006	.09
f-AIMM	10	200	.0024	.01
RWMH	—	—	.25	.25
AMH	—	—	.15	.25
AGM-MH	100	—	.22	.25
IM	—	—	.03	.20

6 Discussion

The starting point of this work is to remark that the information conveyed by the importance weights ratio, which assesses the discrepancy between the target distribution and an independence proposal, although implicitly calculated in an independence M-H transition is lost because of the threshold set to one in the M-H acceptance probability. Indeed, while at X_n , both situations $\{W_n(\tilde{X}_{n+1}) > W_n(X_n), \tilde{X}_{n+1} \sim Q_n\}$ and $\{W_n(\tilde{X}_{n+1}) \gg W_n(X_n), \tilde{X}_{n+1} \sim Q_n\}$ would result in the same transition, regardless of the magnitude difference between $W_n(\tilde{X}_{n+1})$ and $W_n(X_n)$, *i.e.* \tilde{X}_{n+1} will be set as the new state of the Markov chain with probability one. The importance weights are therefore not used to design a relevant adaptation scheme. The core idea of our approach is to retrieve this information by typically incrementing the independence M-H proposal distribution in the later case and not in the former.

We have introduced AIMM, a novel adaptive Markov chain Monte Carlo method to sample from challenging distributions. We show theoretically that this algorithm samples from the ergodic distribution of interest and we illustrate its performance in a variety of challenging sampling scenarios. Compared to other existing adaptive MCMC methods, AIMM needs less prior knowledge of the target. Its strategy of incrementing an initial naive proposal distribution with Gaussian kernels leads to a fully adaptive exploration of the state space. Conversely, we have shown that in some examples the adaptiveness of some other MCMC

samplers may be compromised when an unwise choice of parametric family for the proposal kernel is made. The performance of AIMM depends strongly on the threshold W^* which controls the adaptation rate. This parameter should be set according to the computational budget available. Based on our simulations, AIMM consistently yielded the best tradeoff between fast convergence and low variance.

The adaptive design of AIMM was inspired by Incremental Mixture Importance Sampling (IMIS) (Raftery and Bao, 2010). IMIS iteratively samples and weights particles according to a sequence of importance distributions that adapt over time. The adaptation strategy is similar to that in AIMM: given a population of weighted particles, the next batch of particles is simulated by a Gaussian kernel centered at the particle having the largest importance weight. Even though IMIS and AIMM are structurally different and comparing them is difficult, the computational efficiency of IMIS suffers from the fact that, at each iteration, the whole population of particles must be reweighted in order to maintain the consistency of the importance sampling estimator. By contrast, at each transition, AIMM evaluates only the importance weight of the proposed new state. However, since IMIS calculates the importance weight of large batches of particles, it acquires a knowledge of the state space much quicker than AIMM which accepts/rejects one particle at a time.

We therefore expect AIMM to be more efficient in situations where the exploration of the state space requires a large number of increments of the proposal and IMIS to be more efficient for short run times. To substantiate this expectation, we have compared the performance of AIMM and IMIS on the last example of Section 5 in dimension 4. Figure 12 reports the estimation of the probability $\pi_4(X_1 < -2)$ obtained through both methods for different run times. For short run times IMIS benefits from using batches of particles and gains a lot of information on π_4 in a few iterations. On the other hand, AIMM provides a more accurate estimation of $\pi_4(X_1 < -2)$ after about 150 seconds. Figure 13 illustrates the outcome of AIMM and IMIS after running them for 2,000 seconds. The mixture of incremental kernels obtained by AIMM is visually more appealing than the sequence of proposals derived by IMIS, reinforcing the results from Figure 12.

AIMM can be regarded as a transformation of IMIS, a particle-based inference method, into an adaptive Markov chain. This transformation could be applied to other adaptive importance sampling methods, thus designing Markov chains that might be more efficient than

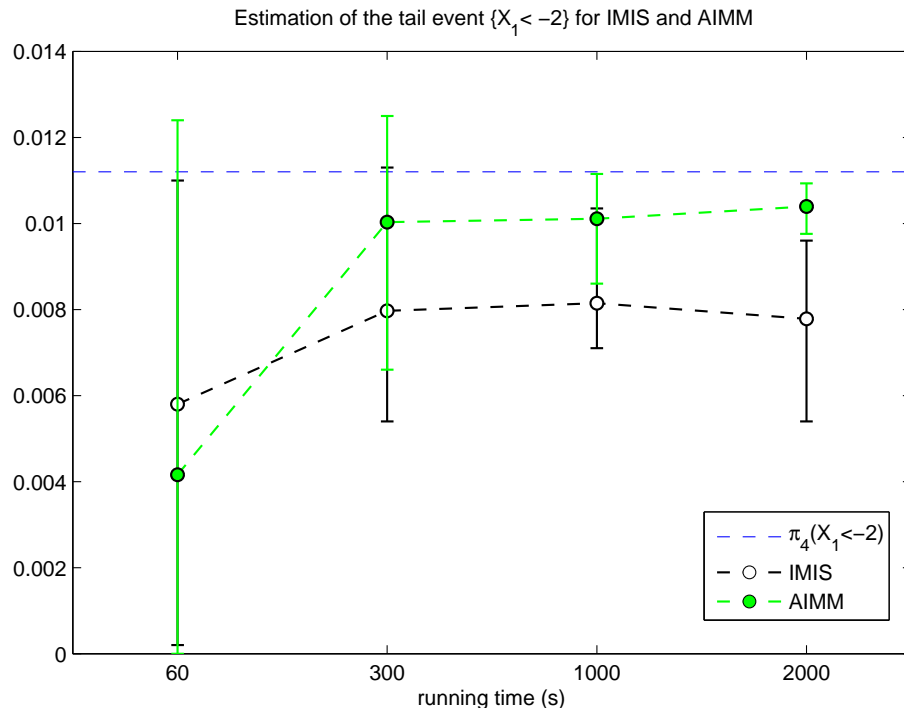


Figure 12: π_4 target, $d = 4$ - Estimation of the probability $\pi_4(X_1 < -2)$ (true value, 0.0112) obtained through AIMM and IMIS for different runtimes (60; 300; 1, 000; 2, 000, in seconds).

their importance sampling counterparts. In a Bayesian context, AIMM could, in addition of sampling the posterior distribution, be used to estimate intractable normalizing constants and evidences via importance sampling. Indeed, the incremental mixture proposal produced by AIMM is an appealing importance distribution since it approximates the posterior and is straightforward to sample from.

Acknowledgements

The Insight Centre for Data Analytics is supported by Science Foundation Ireland under Grant Number SFI/12/RC/2289. Nial Friel's research was also supported by a Science Foundation Ireland grant: 12/IP/1424. Adrian Raftery's research was supported by NIH grants R01 HD054511 and R01 HD070936, and by Science Foundation Ireland grant: 11/W.1/12079. Antonietta Mira's research was supported by Swiss National Science Foundation grant.

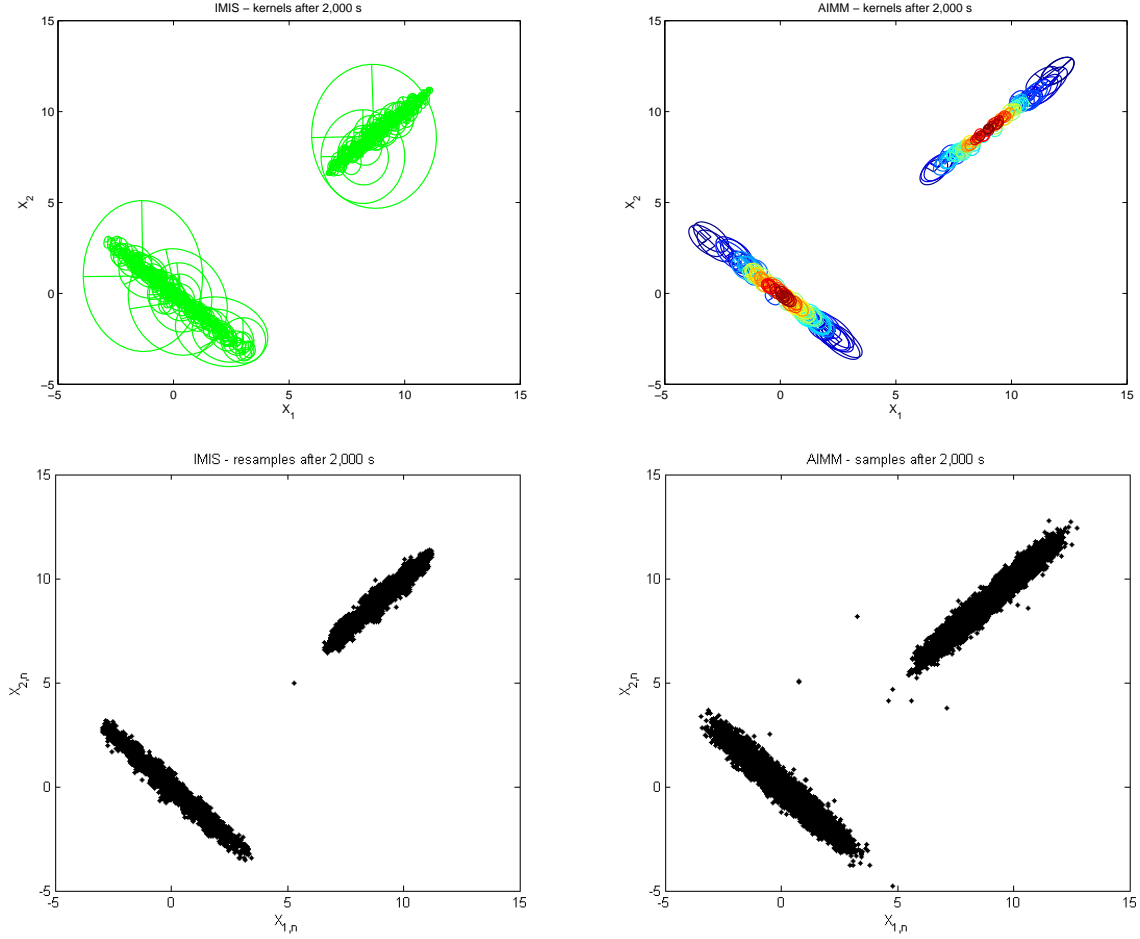


Figure 13: π_4 target, $d = 4$, runtime of 2,000 seconds - Top: Gaussian kernels created by IMIS and AIMM (colours stand for weights assigned to the kernels and the region inside the ellipses contains 75% of each Gaussian mass). Bottom: resamples from IMIS (using the incremental proposal obtained after 2,000 seconds) and samples of the Markov chain simulated through AIMM.

References

- Andrieu, C. and É. Moulines (2006). On the ergodicity properties of some adaptive MCMC algorithms. *Annals of Applied Probability* 16, 1462–1505.
- Gilks, W. R., G. O. Roberts, and S. K. Sahu (1998). Adaptive Markov chain Monte Carlo through regeneration. *Journal of the American Statistical Association* 93, 1045–1054.
- Gilks, W. R. and P. Wild (1992). Adaptive rejection sampling for Gibbs sampling. *Applied Statistics* 41, 337–348.
- Giordani, P. and R. Kohn (2010). Adaptive independent Metropolis–Hastings by fast estimation of mixtures of normals. *Journal of Computational and Graphical Statistics* 19, 243–259.
- Haario, H., E. Saksman, and J. Tamminen (1999). Adaptive proposal distribution for random walk Metropolis algorithm. *Computational Statistics* 14, 375–396.
- Haario, H., E. Saksman, and J. Tamminen (2001). An adaptive Metropolis algorithm. *Bernoulli* 7, 223–242.
- Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* 57, 97–109.
- Hesterberg, T. (1995). Weighted average importance sampling and defensive mixture distributions. *Technometrics* 37, 185–194.
- Holden, L., R. Hauge, and M. Holden (2009). Adaptive independent Metropolis-Hastings. *Annals of Applied Probability* 19, 395–413.
- Liu, J. S. (1996). Metropolized independent sampling with comparisons to rejection sampling and importance sampling. *Statistics and Computing* 6, 113–119.
- Luengo, D. and L. Martino (2013). Fully adaptive Gaussian mixture Metropolis-Hastings algorithm. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6148–6152.

- Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics* 21, 1087–1092.
- Mira, A. (2001). Ordering and improving the performance of Monte Carlo Markov chains. *Statistical Science*, 340–350.
- Pasarica, C. and A. Gelman (2010). Adaptively scaling the Metropolis algorithm using expected squared jumped distance. *Statistica Sinica* 20, 343–364.
- Raftery, A. E. and L. Bao (2010). Estimating and projecting trends in HIV/AIDS generalized epidemics using incremental mixture importance sampling. *Biometrics* 66, 1162–1173.
- Roberts, G. O. and J. S. Rosenthal (2001). Optimal scaling for various Metropolis-Hastings algorithms. *Statistical Science* 16, 351–367.
- Roberts, G. O. and J. S. Rosenthal (2007). Coupling and ergodicity of adaptive Markov chain Monte Carlo algorithms. *Journal of Applied Probability* 44, 458–475.
- Roberts, G. O. and J. S. Rosenthal (2009). Examples of adaptive MCMC. *Journal of Computational and Graphical Statistics* 18, 349–367.
- Roberts, G. O. and J. S. Rosenthal (2011). Quantitative non-geometric convergence bounds for independence samplers. *Methodology and Computing in Applied Probability* 13, 391–403.
- Tran, M.-N., M. K. Pitt, and R. Kohn (2014). Adaptive Metropolis–Hastings sampling using reversible dependent mixture proposals. *Statistics and Computing* 26, 1–21.

A Proof of Proposition 1

Proof. Without loss of generality, we assume that $M_n \equiv n$. The core idea of the proof is to partition the state space \mathbf{X} between the region \mathbf{E}_n^λ where adding the component ϕ_{n+1} yields to a local KL reduction and its complementary subset $\bar{\mathbf{E}}_n^\lambda$. For $\lambda > 0$, define $\mathbf{E}_n^\lambda \in \mathcal{X}$ as the set:

$$\mathbf{E}_n^\lambda = \left\{ x \in \mathbf{X}, \quad (1 + \lambda) \frac{\pi(x)}{\tilde{Q}_{n+1}(x)} \leq \frac{\pi(x)}{\tilde{Q}_n(x)} \right\}. \quad (7)$$

Note that for some $\lambda > 0$, \mathbf{E}_n^λ can be the empty set, but provided that λ is small enough, we are sure that \mathbf{E}_n^λ is non-empty. This results from the construction of ϕ_{n+1} whose probability mass is concentrated on a subset $A \in \mathcal{X}$ where $\tilde{\mathbf{KL}}_n(A)$ is large and which therefore guarantees that $\tilde{\mathbf{KL}}_{n+1}(A) < \tilde{\mathbf{KL}}_n(A)$. By straightforward algebra, we have:

$$\tilde{\mathbf{KL}}_{n+1}(\mathbf{E}_n^\lambda) \leq \tilde{\mathbf{KL}}_n(\mathbf{E}_n^\lambda) - \log(1 + \lambda)\pi(\mathbf{E}_n^\lambda). \quad (8)$$

It can be readily checked that for any $A \in \mathcal{X}$, Jensen's inequality yields

$$\tilde{\mathbf{KL}}_{n+1}(A) - \tilde{\mathbf{KL}}_n(A) \leq \pi(A) \log \left(1 + \frac{\beta_{n+1}}{\bar{\beta}_{n+1}} \right) - \int_A \pi(dx) \frac{\beta_{n+1}\phi_{n+1}(x)}{\sum_{\ell=1}^{n+1} \beta_\ell \phi_\ell(x)}. \quad (9)$$

Combining (8) and (9) applied to $A = \bar{\mathbf{E}}_n^\lambda$, we have $\tilde{\mathbf{KL}}_{n+1} - \tilde{\mathbf{KL}}_n \leq \Delta_n$, where for all $n \in \mathbb{N}$:

$$\Delta_n = \pi(\bar{\mathbf{E}}_n^\lambda) \log \left(1 + \frac{\beta_{n+1}}{\bar{\beta}_{n+1}} \right) - \log(1 + \lambda)\pi(\mathbf{E}_n^\lambda) - \int_{\bar{\mathbf{E}}_n^\lambda} \pi(dx) \frac{\beta_{n+1}\phi_{n+1}(x)}{\sum_{\ell=1}^{n+1} \beta_\ell \phi_\ell(x)}. \quad (10)$$

Note that for all $x \in \mathbf{X}$ and all $\ell \leq n+1$, $\phi_\ell(x) \leq \beta_\ell^{1/\gamma}$ and along with using the log inequality, we have:

$$\Delta_n \leq \pi(\bar{\mathbf{E}}_n^\lambda) \left\{ \frac{\beta_{n+1}}{\bar{\beta}_{n+1}} - \log(1 + \lambda) \frac{\pi(\mathbf{E}_n^\lambda)}{1 - \pi(\mathbf{E}_n^\lambda)} - \frac{\beta_{n+1}}{\sum_{\ell=1}^{n+1} \beta_\ell^{1+1/\gamma}} \mathbb{E}_n^{(\lambda)} \{ \phi_{n+1}(X) \} \right\},$$

where $\mathbb{E}_n^{(\lambda)}$ is the expectation taken with respect to the probability $\pi|_{\mathbf{E}_n^\lambda}$, i.e π restricted to \mathbf{E}_n^λ . Clearly,

$$\mathbb{P}_n(\tilde{D}_n \leq 0) \geq \mathbb{P}_n \left\{ \frac{\beta_{n+1}}{\bar{\beta}_{n+1}} - \log(1 + \lambda) \frac{\pi(\mathbf{E}_n^\lambda)}{1 - \pi(\mathbf{E}_n^\lambda)} - \frac{\beta_{n+1}}{\sum_{\ell=1}^{n+1} \beta_\ell^{1+1/\gamma}} \mathbb{E}_n^{(\lambda)} \{ \phi_{n+1}(X) \} \leq 0 \right\}. \quad (11)$$

Now, let us consider the case $\lambda = 0$. Note that the existence of \mathbf{E}_n^0 with $\pi(\mathbf{E}_n^0) > 0$ is guaranteed under mild regularity assumptions of π . Since we are considering target distributions that are absolutely continuous with respect to the Lebesgue measure, we will admit it. Moreover, note that for all $\lambda > 0$, $\bar{\mathbf{E}}_n^\lambda \supseteq \bar{\mathbf{E}}_0$. With $\lambda = 0$, (11) can now be written as

$$\begin{aligned} \mathbb{P}_n \left\{ \tilde{D}_n \leq 0 \right\} &\geq \mathbb{P}_n \left\{ \frac{\beta_{n+1}}{\bar{\beta}_{n+1}} - \frac{\beta_{n+1}}{\sum_{\ell=1}^{n+1} \beta_\ell^{1+1/\gamma}} \mathbb{E}_n^{(0)} \{ \phi_{n+1}(X) \} \leq 0 \right\}, \\ &= \mathbb{P}_n \left\{ \frac{\sum_{\ell=1}^{n+1} \beta_\ell^{1+1/\gamma}}{\bar{\beta}_{n+1}} \leq \mathbb{E}_n^{(0)} \{ \phi_{n+1}(X) \} \right\}. \end{aligned} \quad (12)$$

On the one hand, provided that $\gamma > 0$, the random variable $\sum_{\ell=1}^{n+1} \beta_\ell^{1+1/\gamma} / \bar{\beta}_{n+1}$ will have limited variability, especially when $\gamma \nearrow 1$. On the other hand, the right hand side can be bounded by below by an increasing quantity. Let $\mathcal{W}_n \in \mathcal{X}$ be the set defined as:

$$\mathcal{W}_n = \{x \in \mathbf{X}, \quad W_n(x) \geq W^*\}. \quad (13)$$

We have:

$$\begin{aligned} \mathbb{E}_n^{(0)} \{ \phi_{n+1}(X) \} &\geq \frac{1}{\pi(\mathbf{E}_n^0)} \int_{\mathcal{W}_n \cap \mathbf{E}_n^0} \phi_{n+1}(x) W_n(x) dQ_n(x) \\ &\geq \frac{W^*}{\pi(\mathbf{E}_n^0)} \int_{\mathcal{W}_n \cap \mathbf{E}_n^0} \phi_{n+1}(x) dQ_n(x). \end{aligned} \quad (14)$$

First, note that for a large enough threshold W^* , the event in the second line of (12) will always hold, and therefore $\mathbb{P}(\tilde{D}_n < 0) \approx 1$. Then, most of the time we will have $\mathbf{E}_n^0 \subset \mathcal{W}_n$, especially when n increases as the adaptation becomes more and more local. Now, (i) since \mathbf{E}_n^0 contains the high density region of ϕ_{n+1} and (ii) $Q_n(\mathbf{E}_n^0) > 0$ as a move $X_{n+1} \sim Q_n$ has been proposed at this location, the integral seems rather constant through the algorithm. However, as n increases $\pi(\mathbf{E}_n^0)$ shrinks as the high density regions of π become well supported by the incremental mixture and therefore the inequality becomes more and more likely.

To prove (ii), we want to show that for all $\varepsilon > 0$,

$$\lim_{n \rightarrow \infty} \mathbb{P}_n(\tilde{\mathbf{K}}\mathbf{L}_{n+1} - \tilde{\mathbf{K}}\mathbf{L}_n < \varepsilon) = 1.$$

Note that for all $\varepsilon > 0$, $\mathbb{P}_n(\tilde{\mathbf{K}}\mathbf{L}_{n+1} - \tilde{\mathbf{K}}\mathbf{L}_n < \varepsilon) \geq \mathbb{P}_n(\Delta_n < \varepsilon) = 1 - \mathbb{P}_n(\Delta_n > \varepsilon)$ and from (10), we have:

$$\lim_{n \rightarrow \infty} \Delta_n \leq -\log(1 + \lambda)\pi(\mathbf{E}_n^\lambda) < 0.$$

The proof follows from noting that $\{\Delta_n > \epsilon\} \rightarrow \emptyset$. □

B Proof of Corollary 1

Proof. The first step of the proof is to show that $D_n - \tilde{D}_n \rightarrow 0$ in probability. Let $g_n : x \mapsto (1/\kappa n)Q_0(x)$, where $\kappa \in (0, 1)$ parameterizes the ω_n (see (3)). Clearly, g_n is a positive function and since Q_0 is bounded, g_n converges pointwise to the null function. Rewrite Q_n as

$$Q_n = (1 - \omega_n) \left(g_n + \tilde{Q}_n \right),$$

it can then be readily checked that, since $\{\omega_n\}_{n>0}$ is a decreasing sequence, we have

$$\frac{Q_{n+1}}{Q_n} \leq \frac{g_{n+1} + \tilde{Q}_{n+1}}{\tilde{Q}_n}. \quad (15)$$

Now considering $|D_n - \tilde{D}_n|$ and Eq. (15), we have:

$$\begin{aligned} D_n - \tilde{D}_n &\leq \int d\pi \log \frac{g_{n+1} + \tilde{Q}_{n+1}}{\tilde{Q}_{n+1}} + \int d\pi \log \frac{\tilde{Q}_n}{g_n + \tilde{Q}_n} \\ &\leq \int d\pi \log \left(1 + \frac{g_{n+1}}{\tilde{Q}_{n+1}} \right) \leq \frac{1}{\kappa n} \int dQ_0 \frac{\pi}{\tilde{Q}_{n+1}}, \end{aligned}$$

where the latter inequality results from the fact that $\log(1+x) \leq x$ for all $x \geq 0$. By the construction of the algorithm, we have that $\text{supp}(Q_0) \subseteq \text{supp}(Q_1)$ since the first incremental kernel is Gaussian and somewhat interpolates samples from Q_0 . Therefore $\text{supp}(Q_0) \subseteq \text{supp}(Q_n)$ for all n and the expectation on the right hand side is bounded from above. This shows that $D_n - \tilde{D}_n \rightarrow 0$ in probability. The proof is completed by noticing that D_n is the sum of two random variables that converge to zero (\tilde{D}_n and $D_n - \tilde{D}_n$) and therefore converges to zero as well, in probability. \square

C Proof of Corollary 2

Proof. It is a consequence of Corollary 1 and of the following expression of $\text{KL}(Q_n, Q_{n+1})$

$$\text{KL}(Q_n, Q_{n+1}) = \int d\pi \frac{1}{W_n} \log \frac{Q_{n+1}}{Q_n}.$$

Note that, because Q_n are mixture of Gaussians, the density is bounded above. Therefore W_n is null only where the density π goes to zero, that is π -almost nowhere. Thus W_n is π almost surely bounded below, say $\eta \leq W_n$ π -a.e. Let $\epsilon > 0$ and write that from Corollary 1:

$$\mathbb{P}_n \left(\int d\pi \log \frac{Q_{n+1}}{Q_n} > \frac{\epsilon}{\eta} \right) \rightarrow 0.$$

The proof is completed by observing that:

$$\mathbb{P}_n \left(\int d\pi \log \frac{Q_{n+1}}{Q_n} > \frac{\epsilon}{\eta} \right) \geq \mathbb{P}_n (\text{KL}(Q_n, Q_{n+1}) > \epsilon) .$$

□

D Proof of Proposition 2

Proof. We first prove that the diminishing adaptation and containment assumptions hold. The *Diminishing adaptation* follows from Corollary 2. Applying Pinsker's inequality (6) allows us to show that $\{\|Q_{n+1} - Q_n\|\}_{n>0}$ goes to zero in probability. Finally, two Metropolis transition kernels that have independence proposal kernels whose total variation vanishes will satisfy diminishing adaptation.

For the containment assumption we use the following result from (Roberts and Rosenthal, 2011, Theorem 6). For any independent Metropolis kernel K , we have

$$\|K^n(x, \cdot) - \pi\| \leq \mathbb{E} (\{1 - \min(m(x), m(Z))\}^n) , \quad (16)$$

where the expectation is with respect to π and for all $x \in \mathbb{X}$, $m(x) = \int Q(dy) \alpha(x, y)$. We now show that the right hand side goes to zero asymptotically. Note that $m(x) \leq 1/W(x)$ where $W = \pi/Q$. We then have:

$$\{1 - \min(m(x), m(Z))\} \leq \left\{ 1 - \min \left(\frac{1}{W(x)}, m(Z) \right) \right\} .$$

By assumption we have that W_n is π -almost everywhere bounded above, say:

$$W_n(x) \leq \lambda, \pi - \text{a.e.}$$

Then, defining the set $A = \{x \in \mathbb{X}, 1 \leq \lambda m(x)\}$, we have:

$$\begin{aligned} \mathbb{E} (\{1 - \min(m(x), m(Z))\}^n) &\leq \int d\pi(z) \left(1 - \min \left(\frac{1}{\lambda}, m(z) \right) \right)^n \\ &= \int_A d\pi(z) \left(1 - \frac{1}{\lambda} \right)^n + \int_{\mathbb{X} \setminus A} d\pi(z) (1 - m(z))^n . \end{aligned} \quad (17)$$

We conclude the proof by using a dominated convergence theorem to show that the right hand side goes to zeros as n goes to infinity. □

E Simulation details

E.1 Competing algorithms

- Adaptive Metropolis–Hastings (AMH) (Haario et al., 2001): a benchmark adaptive MCMC algorithm with non-independence proposal kernel

$$Q_n(x_n, \cdot) = \begin{cases} \Psi_d(\cdot; x_n, \Sigma_0) & \text{if } n \leq N_0 \\ p\Psi_d(\cdot; x_n, \Sigma_0) + (1-p)\Psi_d(\cdot; x_n, \Sigma_n) & \text{otherwise} \end{cases}$$

where $\Psi_d(\cdot; m, S)$ is the d -dimensional Gaussian distribution with mean m , covariance matrix S , Σ_0 is an initial covariance matrix and N_0 the number of preliminary iterations where no adaptation is made. We consider the version of AMH proposed in Roberts and Rosenthal (2009), in which there is a strictly positive probability ($p = .05$) of proposing through the initial Gaussian random walk. The covariance matrix of the adaptive part of the proposal is written $\Sigma_n = s_d \Gamma_n$, where Γ_n is defined as the empirical covariance matrix from all the previous states of the Markov chain X_1, \dots, X_n , and $s_d = 2.4^2/d$ is a scaling factor. The parameters N_0 and Σ_0 were set to be equal to the corresponding AIMM parameters.

- Adaptive Gaussian Mixture Metropolis–Hastings (AGM) (Luengo and Martino, 2013): an adaptive MCMC with an independence proposal kernel defined as a mixture of Gaussian distributions

$$Q_n = \sum_{\ell=1}^M \omega_{\ell,n} \Psi_d(\cdot; \mu_{\ell,n}, \Lambda_{\ell,n}). \quad (18)$$

In AGM, the number of components M is fixed. The proposal distribution Q_n is parameterised by the vector $\Omega_n = \{\omega_{\ell,n}, \mu_{\ell,n}, \Lambda_{\ell,n}\}_{\ell=1}^M$. Given the new state X_{n+1} of the Markov chain, the next parameter Ω_{n+1} will follow from a deterministic update ; see (Luengo and Martino, 2013, section 4) for more details. In the implementation, we set the number of components M identical to the corresponding AIMM parameter M_{\max} .

- For the sake of comparing with non-adaptive methods, we also include the Independence Sampler (IM) (see *e.g* Liu (1996)) and the random walk Metropolis–Hastings

(RWMH) (see *e.g* Roberts and Rosenthal (2001)) using the Matlab build-in function *mhsample* with default settings.

E.2 Performance indicators

The quality of a sampler is based on its ability to explore the state space fully and quickly without getting trapped in specific states (mixing). The KL divergence between π and the stationary Markov chain distribution (if available) and the *Effective Sample Size* are indicators that allow us to assess those two properties. Related statistics such as the Markov chain *Jumping distance* and the *Acceptance rate* of the sampler are also reported. We also provide the chain *Efficiency*, which penalizes the Effective Sample Size by the computational burden generated by the sampler.

- For a Markov chain $\{X_k, k \leq n\}$, we recall that when sampling from π is feasible, the KL divergence between the target and the chain distribution can be approximated by

$$\text{KL} = \frac{1}{L} \sum_{\ell=1}^L \log \left\{ \frac{\pi(Z_\ell)}{\mathcal{L}(Z_\ell | X_{1:n})} \right\}, \quad Z_\ell \sim \pi, \quad (19)$$

where $\mathcal{L}(\cdot | X_{1:n})$ is the kernel density estimation of the Markov chain (obtained using the routine `ksdensity` provided in Matlab in the default setting). We stress that using approximated values for KL are not an issue here as we are first and foremost interested in comparing different samplers, with lower KL being the better.

- The jumping distance measures how fast the sampler explores the state space, see Pasarica and Gelman (2010). For a Markov chain $\{X_k, k \leq n\}$ it is estimated by:

$$\text{JMP} = \frac{1}{n-1} \sum_{k=1}^{n-1} \|X_{k+1} - X_k\|^2, \quad (20)$$

and the larger the squared distance the better.

- The mixing rate of the chain is classically evaluated with the Effective Sample Size (ESS), which is approximated by

$$\text{ESS} = 1 / \left\{ 1 + 2 \sum_{t=1}^T \hat{\rho}_t \right\}, \quad (21)$$

where $\hat{\rho}_t$ denotes the empirical lag t covariance estimated using the sample path of $X_{1:n}$, $T = \min(1000, t_0)$ and t_0 is the smaller lag such that $\hat{\rho}_{t_0+\ell} < .01$, for all $\ell > 0$. $\text{ESS} \in (0, 1)$ ($\text{ESS}=1$ corresponds to *i.i.d.* samples) and the higher ESS , the better the algorithm. When $d > 1$, ESS is set as the minimum ESS among the d marginal ESS 's.

- The tradeoff between the computational efficiency and the precision is estimated by EFF

$$\text{EFF} = \text{ESS} / \tau, \quad (22)$$

where n is the total number of iterations performed by the sampler and τ the CPU time (in second) required to achieve the N transitions (Tran et al., 2014).

E.3 AGM targeting π_2

Although similar to AIMM, AGM adapts a Gaussian mixture proposal to the target, the distribution of the resulting Markov chain remains far from π_2 after 200,000 iterations. Figure 14 gives a hint to understand why AGM is not efficient in sampling from π_2 . By construction, AGM adapts locally a component of the proposal, provided that some samples are available in its vicinity. As a result, components initially located where the probability mass is non-negligible will adapt well to the target but the others will see their weight shrinking to zero (in dimension two, out of one hundred initial kernels having the same weight, the ten kernels with highest weight hold 0.99 of the total mixture weight after 200,000 iterations). This gives those kernels with initial poor location a high inertia, which prevents them moving away from low density regions to reach regions that are yet to be supported by the proposal. AIMM's opportunistic increment process explores the state space more efficiently.

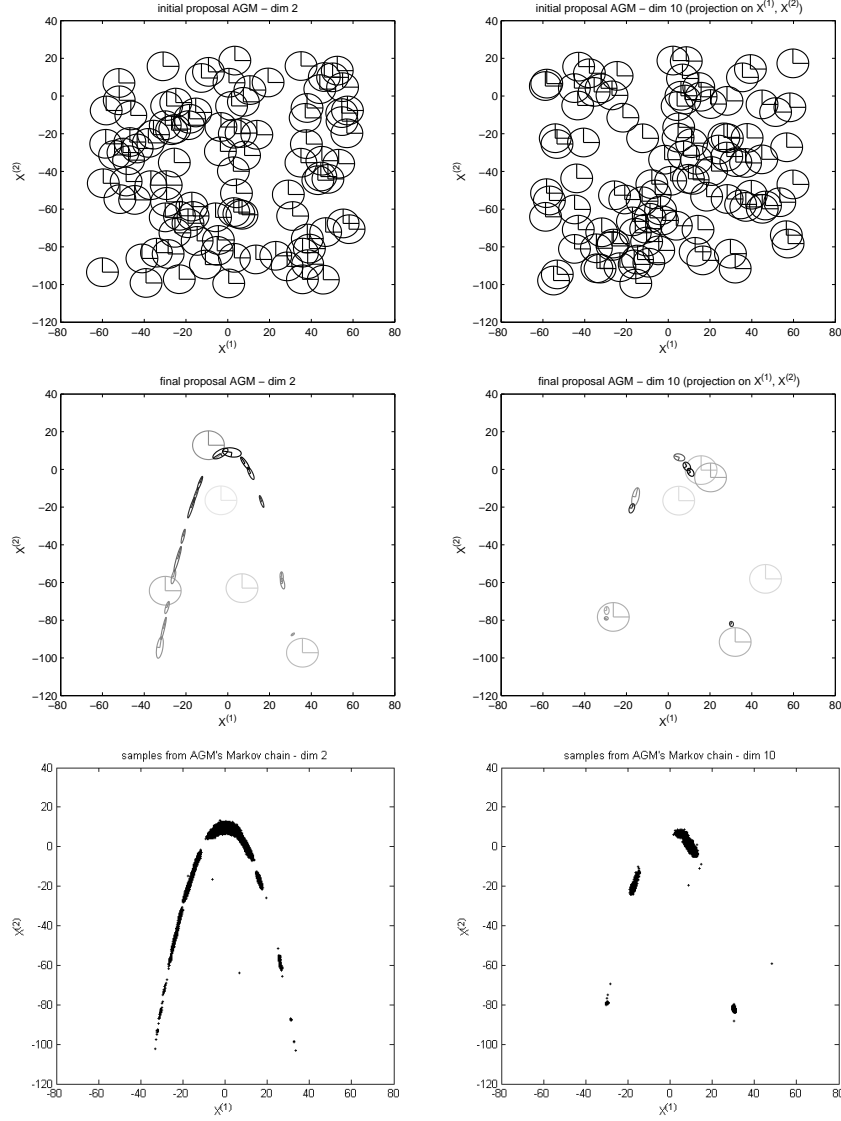


Figure 14: π_2 target, $d \in \{2, 10\}$ - AGM sampling from π_2 in dimensions $d = 2$ (left column) and $d = 10$ (right column, projection onto the first two dimensions). First and second rows: initial mixtures with centers drawn uniformly and adapted mixtures after $n = 200,000$ iterations. The grey levels on the ellipses stand for the weights of the components (from white to black for larger weights) and the region inside the ellipses contains 75% of each Gaussian mass. Third row: samples $\{X_{1,2}, X_{2,n}, n \leq 200,000\}$ from AGM.